MASTER EN SCIENCES INFORMATIQUES

Principes de signalisation aux interfaces UNI et NNI dans un réseau ATM privé

Mottiat, Frédéric

Award date:
1996

Awarding institution:
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année académique 1995-1996

PRINCIPES DE SIGNALISATION AUX INTERFACES UNI ET NNI DANS UN RESEAU ATM PRIVE

Frédéric Mottiat

Promoteur : Monsieur Philippe van Bastelaer

Mémoire présenté en vue de l'obtention du grade de Licencié et Maître en Informatique

CONDENSE

Ce mémoire traite de la signalisation dans les réseaux ATM privés. Après avoir introduit les concepts résidant à la base de la technologie ATM, nous présenterons les couches de protocoles impliquées dans un modèle de signalisation. Nous commencerons notre examen de ce modèle par une étude des protocoles SSCF et SSCOP ayant pour but de fournir à la couche de signalisation une capacité de transfert fiable des informations. Nous continuerons l'examen de notre modèle par l'étude des protocoles UNI 3.1 et PNNI spécifiés par l'ATM Forum et utilisés respectivement pour les procédures de signalisation aux interfaces utilisateur-à-réseau et réseau-à-réseau. L'étude de ces deux protocoles se fera de manière graduelle : nous tenterons en premier lieu de trouver les étapes principales par lesquelles passe toute procédure de signalisation et affinerons au fur et à mesure celles-ci par l'introduction de nouveaux éléments et concepts. Nous conclurons ce mémoire par une définition précise de ce qu'est la signalisation et de ce qu'elle doit permettre.

ABSTRACT

This thesis is dedicated to the study of signaling in private ATM networks. After the concepts that lie at the basis of the ATM technology have been introduced, we will present the protocol layers involved in a signaling model. We will begin the model's examination with a study of the SSCF and SSCOP protocols that offer a reliable transfer capacity of informations to the signaling protocol. We will continue our model's examination with the study of the UNI 3.1 and PNNI protocols used respectively for signaling procedures at the user-to-network and network-to-network interfaces. The study of those latter protocols will be done in a progressive way : we will try to find the major steps every signaling procedure has to go throught and refine those steps with the introduction of new elements and concepts. We will finish this thesis with a precise definition of signaling and what it has to offer.

Remerciements

Je tiens tout particulièrement à remercier Monsieur Ectors sans qui je n'aurais pu réaliser le stage qui est à la base de ce mémoire ainsi que Monsieur Eric Levy qui m'a dirigé durant les 6 mois de stage chez IBM.

Je remercie mon promoteur, Monsieur Philippe van Bastelaer, pour avoir soutenu ce projet dès le début ainsi que pour ses conseils, orientations et corrections.

Pour leur support lexical, syntaxique et sémantique, je remercie ma mère et Monsieur Jean-François Gobbers. J'espère être parvenu, par l'intermédiaire de ce mémoire, à leur faire partager mon intérêt pour les télécommunications et les réseaux.

Je remercie également les membres de la liste de discussion Internet *Cell-Relay* qui m'ont été d'un grand secours dans l'étude quelques fois rébarbative des protocoles exposés dans ce mémoire.

"Ni l'ignorance n'est défaut d'esprit, ni le savoir n'est preuve de génie."

Vauvenargues, *Réflexions et Maximes*, 217 [1746]

Tables des matières

Table des figures

Table des tableaux

0. Introduction

Depuis une décennie, les systèmes informatiques ont connu une remarquable augmentation de puissance, tant au niveau des mémoires et des processeurs que des autres périphériques. L'information traitée n'est plus seulement constituée de texte, mais d'un ensemble de différents supports tant visuels qu'auditifs. Le mot clé des années 1990, et vraisemblablement de la prochaine décennie, est et sera le « multimédia ».

Les années 1990 auront également vu apparaître le phénomène des « autoroutes de l'information ». Qui aujourd'hui, informaticien ou non, n'a jamais entendu parler du réseau Internet, souvent décrit comme précurseur de ce que pourraient offrir ces nouvelles autoroutes ? Les réseaux constituant ces autoroutes de l'information véhiculeront toute sorte d'informations : textes, images, bandes sonores ou vidéos, programmes informatiques ou tout autre support combinant les précédents.

Tout comme les systèmes informatiques, les technologies de réseau évoluent de manière à pouvoir supporter les nouveaux trafics induits par les nouveaux supports multimédias; elles doivent supporter des trafics vocaux, vidéo ou de données "brutes", tous ayant des caractéristiques individuelles résultant en des demandes d'utilisation différentes du canal de communication. Examinons brièvement les différents types de trafic et leurs besoins sur un canal de communication :

- La voix : elle a une génération asynchrone (un interlocuteur peut parler à n'importe quel moment) et une transmission synchrone (une fois le message parti, il doit être transporté de manière continue sur le réseau). Les besoins en largeur de bande sont relativement peu élevés et constants (64 Kbps). Les signaux peuvent contenir un haut degré d'erreur sans que cela porte préjudice à la compréhension de la communication [NORM].
- La vidéo : sa génération est synchrone (continue) et sa transmission est synchrone. Les besoins en largeur de bande peuvent varier de 64 Kbps à plusieurs Mbps, selon la complexité et le mouvement dans l'image. Le contrôle d'erreurs doit être assez sévère afin d'éviter une mauvaise interprétation des données affichées [NORM].
- Les données « brutes » : leur génération peut être synchrone ou asynchrone. La transmission est en général asynchrone (il n'y a pas de relation de temps particulière entre l'émetteur et le récepteur). Les besoins en largeur de bande peuvent varier de quelques Kbps à plusieurs Mbps. Le contrôle d'erreurs doit d'habitude être très serré car ce type d'information est très sensible aux erreurs [NORM].

C'est pourquoi le Comité Consultatif International sur la Téléphonie et Télégraphie (CCITT) a choisi ATM comme technique de commutation et de multiplexage pour base des réseaux Broadband ISDN (B-ISDN) destinés à supporter ces différents types de trafic. La recommandation I.121 du CCITT dit : « *B-ISDN supporte les communications commutées, semi-permanentes et permanentes, en point-à-point ou en point-à-multipoint et fournit sur demande des services réservés et permanents. Les connexions en B-ISDN supportent le mode circuit et le mode paquet de type mono et/ou multimédia en mode connecté ou non connecté, dans une configuration bidirectionnelle ou unidirectionnelle. B-ISDN contiendra des capacités intelligentes afin de pouvoir fournir des caractéristiques de service avancées, supportant des outils puissants d'opération et de maintenance, de contrôle et de gestion des réseaux.* » [STA92]

ATM devient rapidement un standard dans l'industrie et ceci en majeure partie grâce à l'ATM Forum.

L'ATM Forum est une organisation internationale fondée en 1991 dont l'objectif est d'accélérer l'utilisation des produits et services ATM par l'intermédiaire d'une convergence rapide des spécifications d'interopérabilité. Constitué à la base de 4 membres, l'ATM Forum compte à ce jour plus de 750 compagnies et sociétés membres représentant tous les secteurs des communications et de l'industrie

informatique ainsi qu'un certain nombre d'agences gouvernementales, organisations de recherches et autres utilisateurs [FORUM].

Sous l'égide de l'ATM Forum, les parties les plus importantes de la technologie ATM ont déjà été standardisées et d'autres sont sur la même voie. Ainsi, jusqu'à présent, les standards développés en grande majorité par l'ATM Forum ou par l'ITU-T (i.e. l'ancien CCITT) couvrent [FELD95] :

- la spécification de l'interface entre un utilisateur ATM et le réseau ATM auquel il est connecté (appelée UNI pour *User-to-Network Interface*) : standardisation des communications entre un équipement utilisateur - un CPE pour *Customer Premises Equipment* ou TE pour *Terminal Equipment* - et les réseaux de transport;
- la spécification des interfaces d'échange de données (DXI : *Data Exchange Interface*) qui définit comment des bridges, routeurs et hubs d'une technologie autre qu'ATM peuvent agir en tant que "processeurs frontaux" pour des réseaux ATM, facilitant ainsi une transition douce d'une technologie de réseau actuelle vers ATM;
- la spécification des standards d'émulation LAN sur ATM;
- la spécification de protocoles de création de circuits virtuels au travers d'un réseau;
- la spécification de protocoles permettant l'interconnexion de réseaux publics ATM;
- la spécification de protocoles permettant la gestion du trafic et du réseau.

Les réseaux ATM actuels - quoiqu'implémentant la quasi-totalité de ces standards - sont encore relativement lourds du point de vue de leur utilisation. Ceci ne s'applique pas au transfert d'informations lui-même - cet aspect étant déjà fortement standardisé - mais à la création d'une connexion entre utilisateurs distants. ATM est en effet un protocole exclusivement orienté connexion; il est donc nécessaire, avant tout transfert d'informations en provenance des utilisateurs, d'ouvrir une connexion entre eux.

La création d'une connexion entre deux utilisateurs se fait encore dans la plupart des cas par une configuration ou une création manuelle d'un chemin traversant un ou plusieurs réseaux. Si ceci est "supportable" dans des réseaux de faible envergure ou tels que les connexions demandées seront toujours identiques, cette solution n'est plus envisageable pour des réseaux de grande envergure ou tels que les connexions demandées par les utilisateurs ne seront plus prévisibles.

Afin de résoudre ce problème, des protocoles de signalisation ont été spécifiés tant par l'ATM Forum pour les réseaux privés que par l'ITU-T pour les réseaux publics. Nous pourrions, dans un premier temps, définir un protocole de signalisation comme étant ce qui va permettre la création ou la destruction de chemins entre des utilisateurs sur demande de ceux-ci, chemins utilisés pour le transfert d'informations entre ces utilisateurs.

Ce mémoire a pour but d'étudier les protocoles de signalisation définis pour les réseaux ATM privés.

Dans un premier chapitre, nous redéfinirons les concepts de base du protocole ATM. Nous y verrons ce qu'est une cellule ATM - l'unité de base de transport des informations - et comment celle-ci est structurée. Nous verrons que le modèle ATM peut être découpé en couches, comme pour le modèle OSI et que chaque couche joue un rôle bien déterminé afin d'offrir à l'utilisateur le service qu'il a demandé (transfert de données, de son, d'images, demandant ou non une synchronisation temporelle entre les utilisateurs). Le modèle ATM peut également être divisé en plans - notion bien différente que celle de la découpe en couches - où chaque plan se rattache à un aspect particulier des transactions. Ce premier chapitre a principalement pour but de donner au lecteur une connaissance de base sur ATM afin de faciliter la lecture et la compréhension des chapitres suivants. Le lecteur jouissant déjà cette connaissance peut sans soucis débuter la lecture au deuxième chapitre.

L'architecture générale du modèle de signalisation sera exposée dans le deuxième chapitre. Nous y présenterons l'ensemble des couches de protocoles constituant celle-ci. Après avoir redéfini ce qu'est un protocole de signalisation suite aux connaissances que nous aurons acquises dans le premier chapitre, nous tâcherons plus particulièrement d'étudier les protocoles utilisés par un protocole de signalisation et verrons en quoi ceux-ci sont nécessaires : quels sont les services et les fonctionnalités spécifiques qui font que ces protocoles doivent absolument être utilisés ? Qu'offrent-ils à une couche de signalisation ? Telles sont les questions que nous tenterons de résoudre. Les fonctionnalités des différentes couches prenant part au modèle de signalisation étant alors connues, nous conclurons le chapitre par une comparaison directe des couches du modèle de signalisation ATM et du modèle OSI.

Les deux chapitres suivants sont dédiés à l'étude de deux protocoles de signalisation spécifiés par l'ATM Forum. Nous parlons ici de deux protocoles car deux cas de signalisation distincts doivent être envisagés. Premièrement, comment l'utilisateur peut-il demander l'ouverture d'une connexion à son réseau (plus particulièrement : à son point d'accès au réseau) et inversement, comment le point d'accès au réseau d'un utilisateur peut-il avertir celui-ci de l'arrivée d'une demande d'ouverture de connexion le concernant ? Deuxièmement, que se passe-t-il, à l'intérieur du réseau, pour qu'une demande d'ouverture de connexion parvienne jusqu'au point d'accès au réseau de l'utilisateur appelé ? Les mêmes questions peuvent également être posées pour la fermeture d'une connexion.

Dans le troisième chapitre, nous verrons donc comment un utilisateur peut demander à son réseau l'ouverture ou la fermeture d'une connexion avec un utilisateur distant. Par une approche graduelle, nous tenterons de découvrir quelles sont les étapes principales d'une procédure de connexion et descendrons chaque fois d'un degré de détail supplémentaire. Nous terminerons le chapitre par des exemples choisis de procédures de signalisation mettant en pratique les concepts que nous aurons soulevés.

Le quatrième chapitre aura pour objectif d'étudier ce qui se passe au sein même d'un réseau lorsque l'on aborde le problème de la signalisation. Nous verrons entre autres que si l'on parle de signalisation, un autre concept doit obligatoirement être abordé : celui du routage. C'est en effet grâce au routage qu'un chemin menant jusqu'à la destination d'une demande de connexion pourra être trouvé et qu'à partir de ce moment, des procédures de signalisation pourront être mises en œuvre afin de tailler le chemin trouvé au travers du réseau. Cette notion de routage entraîne elle-même la connaissance d'un nombre élevé d'informations sur la topologie du réseau; nous verrons qu'il peut s'avérer nécessaire de construire une vue du réseau autre qu'un simple ensemble plat de commutateurs reliés entre eux par des liens physiques. L'étude de la signalisation dans ce chapitre sera fortement calquée sur le chapitre 3 et, tout comme pour celui-ci, nous terminerons le chapitre par des exemples parlants de procédures de signalisation.

Le cinquième et dernier chapitre, confidentiel, est consacré au stage effectué chez IBM du mois d'août 1995 au mois de janvier 1996. Ce stage a été consacré au développement d'un prototype de signalisation pour un commutateur WAN d'IBM mettant en œuvre les protocoles que nous aurons vu aux deuxième, troisième et quatrième chapitres.

1. ATM

Ce chapitre est consacré à une présentation générale des concepts de base des réseaux de type ATM nécessaires à la compréhension des chapitres suivants. Le lecteur se sentant confortable avec les notions de cellule ATM, de découpe en couches et en plans du modèle ATM, des fonctionnalités associées à ces couches et plans et des notions d'adressage peut directement passer au chapitre deux.

1.1 Bases d'ATM

Les sections suivantes n'ont pas pour but de constituer une présentation exhaustive d'ATM, mais bien une introduction aux concepts de base de cette technologie de réseau. Le lecteur désireux d'approfondir son étude sur ATM se référera à l'abondante littérature couvrant ce sujet. Des propositions de lectures sont données en fin de chapitre.

1.1.1 Format de cellules

Dans les réseaux ATM, l'information est transportée dans des paquets de taille fixe appelés cellules. Afin de pouvoir supporter tous les types de trafic définis ci-dessus (voix, vidéo, données), on a choisi de prendre des cellules de très petite taille. De cette manière, le délai induit par chacun des paquets sera court et probablement fixé, permettant à un trafic vocal ou vidéo d'être mélangé à un trafic de données sans diminution de la qualité de réception (les cellules se référant à des trafics différents peuvent et sont en général multiplexées sur un même support physique). La taille des cellules a été fixée à 53 octets répartis en un champ de 48 octets destinés aux informations de l'utilisateur et un en-tête de 5 octets destiné à transporter des informations de routage. Cet en-tête ne contient pas une adresse explicite mais un label, l'adressage explicite n'étant pas possible à cause de la taille courte et fixe des cellules. Lors de l'établissement du protocole ATM par le CCITT, celui-ci jugea qu'un overhead de 10 % par rapport à la charge utile de la cellule (les 48 octets utilisateur) était une borne supérieure maximale, d'où le choix d'un en-tête de 5 octets. L'adresse ATM dans les réseaux privés comptant 20 octets en taille, celle-ci n'aurait pu être insérée dans l'en-tête.

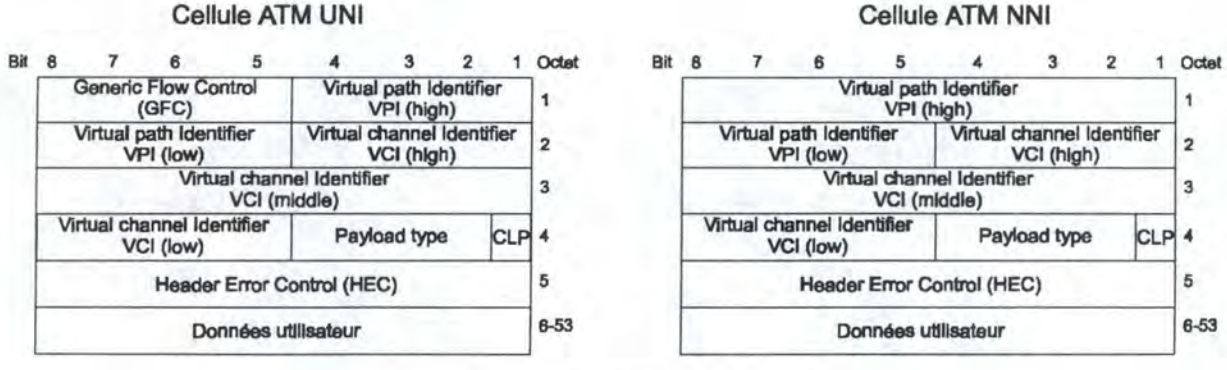


Figure 1-1: cellules ATM UNI et NNI

La Figure 1-1 illustre deux formats de cellules ATM. Le premier format, appelé cellule ATM UNI pour User-to-Network Interface, correspond aux cellules allant de la station de l'utilisateur (ou d'une autre ressource ATM telle qu'un commutateur LAN) à son point d'entrée au réseau (i.e. un commutateur ATM). Le deuxième format de cellule, le format NNI pour Network-to-Node Interface, correspond aux cellules transportées entre deux commutateurs ATM. La seule et unique différence entre ces deux formats de cellules se situe dans la présence d'un contrôle générique de flux (GFC) dans les cellules

ATM UNI. Celui-ci n'a qu'une signification locale à l'interface entre l'utilisateur et le réseau et est utilisé pour différencier deux modes d'opération :

1. *Mode contrôlé* : une valeur est donnée au GFC afin d'assurer un contrôle de flux sur les cellules. Cependant aucun format précis n'a été défini jusqu'ici par les organismes de standardisation.
2. *Mode non contrôlé* : l'ensemble des bits du GFC sont mis à zéro.

Les éléments suivants sont communs aux deux formats de cellules :

1. Virtual Channel Identifier (VCI) : l'identificateur d'un canal virtuel. Un canal virtuel définit une route de transmission unidirectionnelle utilisée pour le transport des cellules. Ce champ a une longueur de 16 bits, définissant donc 65536 canaux différents.
2. Virtual Path Identifier (VPI) : l'identificateur d'un chemin virtuel. Un chemin virtuel est constitué d'un ensemble de canaux virtuels. A chaque identificateur d'un chemin virtuel correspond donc un ensemble de canaux virtuels. Comme le montre la Figure 1-1, le champ VPI d'une cellule NNI est de 4 bits plus grand que celui d'une cellule UNI, vu l'allocation des 4 bits du champ de GFC au champ VPI dans les cellules NNI. Ceci est dû au fait que le nombre de chemins virtuels utilisés entre nœuds de réseau est supposé être plus grand que le nombre de chemins virtuels entre une ressource ATM et le nœud auquel elle est attachée. On a donc 256 chemins virtuels au maximum à l'interface entre utilisateur et réseau et 4096 chemins virtuels au maximum entre nœuds de réseau. On se reportera à la section 1.1.4 "Notion de VPI, VCI et de commutation" pour une explication sur la notion de chemin virtuel.
3. *Header Error Control* (HEC) : champ utilisé pour valider l'en-tête de cellule. On remarquera donc que le contrôle d'erreurs se fait ici seulement sur l'en-tête de cellule et non sur son contenu. On trouvera une description de l'algorithme utilisé pour effectuer ce contrôle d'erreurs en annexe A.
4. *Cell Loss Priority* (CLP) : on distingue deux types de cellule (UNI et NNI confondus) : les cellules à haute priorité (valeur 0) et les cellules à basse priorité (valeur 1). En cas de saturation d'un lien entre deux commutateurs, les cellules marquées CLP = 1 seront rejetées par préférence.
5. *Payload* Type (PT) : ce champ est utilisé afin de différencier les cellules contenant des informations en provenance de l'utilisateur de cellules non utilisateur. On trouvera une définition des différents types de cellule par payload type dans [KYAS95], page 125.

1.1.2 Modèle en couches

La découpe du modèle ATM se fait en couches, comme l'a introduit l'ISO avec le modèle OSI. La découpe en couches se fait par fonctionnalité logique. Comme le montre la Figure 1-2, il est tentant de comparer directement le modèle OSI et le modèle ATM et ainsi d'attribuer les fonctionnalités des couches physiques, logiques, réseaux et supérieures du modèle OSI aux couches physiques, ATM et ATM Adaptation Layer (AAL) du modèle ATM. Ce raisonnement serait toutefois incorrect. Dans les sous-sections suivantes, nous décrirons les fonctionnalités relatives à chacune des couches du modèle ATM. Dans le chapitre suivant,

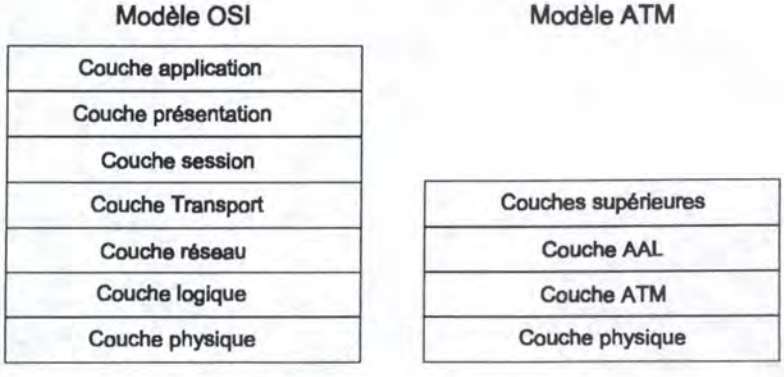


Figure 1-2 : modèle en couches OSI et ATM

nous tenterons d'établir un parallèle entre les couches du modèle OSI et les couches du modèle ATM à la section 2.4.

1.1.2.a) Couche physique

Le rôle de la couche physique est de transformer le flux de cellules à émettre provenant des couches supérieures en un flot de bits en utilisant le support de transmission disponible et inversement de transformer le flot de bits en un flux de cellules pour les couches supérieures. Cette couche est divisée en deux sous-couches : la sous-couche Transmission Convergence (TC) et la sous-couche Physical Medium.

Les rôles principaux associés à la sous-couche Transmission Convergence sont :

- la génération et la vérification du HEC de chaque cellule;
- le *Cell Delineation* : les cellules peuvent, selon le support physique qui aura été choisi, être transportées dans des trames de transmission, chaque trame transportant plusieurs cellules. Lors de la réception des premières trames, la couche physique effectue une phase de synchronisation permettant de trouver les limites de chaque cellule dans ces trames. Cette phase peut être décrite comme suit : lorsqu'une trame est reçue la couche physique tente de trouver une corrélation entre les 4 premiers octets de ce qu'elle pense être une cellule ATM et le $5^{ème}$ octet contenant normalement le HEC. Si cette corrélation est trouvée alors les 48 octets suivant constituent le payload de la cellule (i.e. le contenu utilisateur de 48 octets). Après que ce test ait été effectué un certain nombre de fois la couche physique se déclare "synchronisée" et cette fonction n'est plus utilisée.
- l'encapsulation des cellules dans les trames de transmission du média de transport utilisé à l'émission et l'extraction des cellules de ces trames à la réception.

Les rôles principaux associés à la sous-couche Physical Medium sont :

- la transmission/réception d'un flot continu de bits accompagnés d'information de timing nécessaire à la synchronisation de l'émission/réception.

Les supports physiques les plus couramment utilisés dans les réseaux ATM sont le câble coaxial (75 ohms) et la fibre optique. Cependant, pratiquement tout support physique, y compris la paire torsadée, peut être utilisé dans les réseaux de type ATM. Tout ce qui est nécessaire, c'est une sous-couche TC adaptée à ce support. Notons qu'Olivetti a effectué des tests positifs dans l'utilisation d'ATM par radio, avec une vitesse de transmission de 10 Mbits/s.

La Figure 1-3 reprend l'ensemble des supports physiques supportés actuellement par ATM. L'ATM Forum a émis des spécifications pour tous les types d'interface présentés dans cette figure, à l'exception du transfert sans fils [BLA95].

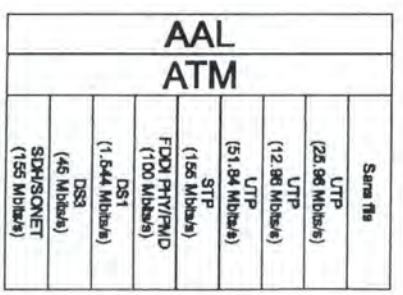


Figure 1-3 : supports physiques supportés par ATM

1.1.2.b) Couche ATM

Cette couche est indépendante du type de support physique utilisé. Ses rôles principaux sont :

- la génération des informations de contrôle de flux : des informations de contrôle de flux sont introduites dans l'en-tête des cellules. Ceci n'est donc valide que pour les cellules entre utilisateur/ressource ATM et nœud réseau auquel il est connecté;
- le multiplexage et démultiplexage des cellules : des cellules appartenant à des *virtual channels* (VC) ou *virtual paths* (VP) différents mais utilisant le même support physique sont multiplexées selon la méthode TDM (Time-Division Multiplexing);
- la commutation et traduction des VP/VC : la couche ATM a pour rôle de commuter les cellules sur base des valeurs VPI et VCI contenues dans l'en-tête de celles-ci. Pendant la commutation des cellules dans un nœud du réseau il est nécessaire de traduire un VPI et/ou VCI en un autre ensemble de valeurs, ces valeurs étant uniquement locales à un lien entre deux nœuds;
- la génération des cellules : les données à envoyer sont fournies à la couche ATM par les couches supérieures sous forme de paquets de 48 octets. La couche ATM fabrique alors les cellules ATM contenant ces données en y ajoutant l'en-tête de cellule. Cependant le champ HEC de l'en-tête est généré par la couche physique;
- le *Cell Rate Decoupling* : des cellules vides sont introduites afin de s'adapter à la capacité (largeur de bande) du support utilisé. Le rôle de cette fonction est de pouvoir assurer un synchronisme avec la vitesse de transmission du support utilisé. Les cellules vides ne sont pas passées aux couches supérieures. Notons que ce procédé n'est nécessaire que pour les systèmes physiques de transmission utilisant des time-slots synchrones pour le transfert des cellules. [KYAS95]

On se reportera à la section 1.1.4 *"Notion de VPI, VCI et de commutation"* pour une bonne compréhension des notions de canaux et chemins virtuels et des différents modes de commutation.

1.1.2.c) Couche AAL

Un réseau ATM peut supporter plusieurs types de trafic différents (données, voix, vidéo en service interactif - vidéoconférence par exemple - ou en simple consultation - vidéo sur demande). La sensibilité de ces différents types d'application à des problèmes de déformation ou de perte des cellules, ou encore à des problèmes de congestion sur le réseau est différente.

On peut isoler trois critères majeurs permettant de différencier les besoins de ces différents types d'application sur le réseau :

1. L'application est-elle sensible à la synchronisation temporelle entre émetteur et récepteur ? Ceci est le cas pour des applications telles que l'émulation de circuit ou la vidéo.
2. L'application utilise-t-elle un débit binaire constant ou variable ?
3. L'application nécessite-t-elle un fonctionnement en mode connecté ou non connecté ?

Selon la combinaison des trois critères donnés ci-dessus, on distingue cinq classes de services différentes, comme illustré au Tableau 1-1.

	Classe A	*Classe B*	*Classe C*	*Classe D*	*Classe X*
Synchronisation temporelle	Requise		Non Requise		Défini par l'utilisateur
Débit binaire	Constant	Variable			Défini par l'utilisateur
Mode de connexion	Mode connecté			Mode non connecté	Défini par l'utilisateur
Exemple	Emulation de circuit	Vidéo	Transfert de données orienté connexion	Transfert de données non orienté connexion	Défini par l'utilisateur
Définition qualitative	Constant Bit Rate (CBR)	Variable Bit Rate (VBR) en temps réel	VBR en temps non réel	VBR en temps non réel	User Bit Rate (UBR)

Tableau 1-1: classes de services offerts par la couche AAL

Les définitions qualitatives données au Tableau 1-1 sont une autre manière d'identifier une classe de service.

UBR est un service particulier défini pour des applications non sensibles au délai de transmission ou à la synchronisation temporelle et ne nécessite aucune réservation préalable de largeur de bande, contrairement à CBR et VBR. Il est généralement utilisé pour des applications de communication "traditionnelles" entre ordinateurs.

La couche d'adaptation ATM a pour but d'offrir les différents services exposés au Tableau 1-1 aux applications des couches supérieures. Elle offre à celles-ci un service de transmission fiable, assurant le contrôle de flux et d'erreur si celui-ci est requis ainsi que la synchronisation temporelle lorsque celle-ci est nécessaire.

Le transfert d'informations propres aux couches AAL ne se fait qu'entre les deux entités AAL distantes (entre les deux utilisateurs connectés à travers le réseau), comme illustré par la Figure 1-4. Les nœuds de réseau ne sont pas concernés par les informations relatives aux couches AAL et n'implémentent donc pas cette couche. En effet le rôle des nœuds pendant le transfert d'informations est uniquement d'agir comme commutateurs et donc seule la couche ATM est concernée.

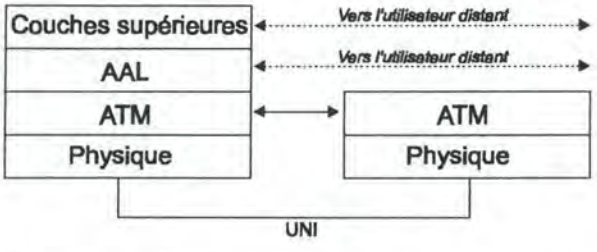


Figure 1-4 : relation entre entités paires pour les couches AAL

On distingue cependant un dernier type de couche AAL : la couche S-AAL, pour *Signalling ATM Adaptation Layer*. Cette couche est utilisée dans les phases d'ouverture / fermeture de connexion et est cette fois implémentée tant dans les équipements utilisateurs que dans les nœuds du réseau. Cette couche particulière sera exposée au chapitre suivant.

On définit 4 types de couche AAL différents, selon les besoins requis : AAL type 1, type 2, type 3/4 et type 5. De manière générale, AAL type 1 a été défini pour la classe de service A, le type 2 pour la classe B, le type 3/4 pour la classe C et le type 5 pour la classe D. Cependant, cette association n'est pas unique et fixe : AAL 3/4 peut offrir aussi bien des services en mode connecté qu'en mode non connecté,

de même pour AAL 5. AAL5 est une simplification de AAL 3/4 et est utilisé principalement pour des applications particulières de niveau supérieur telles que l'émulation LAN et IP sur ATM et pour les protocoles de signalisation. Ajoutons également que s'il existe des spécifications et recommandations de l'ITU-T pour les types d'AAL 1, 3/4, 5 et défini par l'utilisateur (correspondant à la classe X), la classe 2 par contre est toujours l'objet d'études et n'a pas encore de recommandation associée.

Les différents types de couche AAL sont subdivisés en deux sous-couches : la sous-couche *convergence* et la sous-couche "segmentation et réassemblage" (SAR : *Segmentation and Reassembly*), cette dernière se trouvant en dessous de la sous-couche *convergence*.

La sous-couche *convergence* définit le type de service offert par la couche AAL aux couches supérieures. Selon le type de service demandé, elle assure la correction d'erreurs et le contrôle de flux pour les applications sensibles aux pertes de données (AAL type 3/4 et 5) et la synchronisation entre émetteur et récepteur pour les applications sensibles au temps (AAL type 1 et 2).

La sous-couche SAR est responsable de la segmentation des informations provenant des couches supérieures en paquets de taille appropriée (48 octets) pour la construction des cellules par la couche ATM. Inversement, elle réassemble les paquets fournis par la couche ATM en informations destinées aux couches supérieures.

1.1.3 Modèle en plans

Outre la découpe en couches du modèle ATM, il existe une autre découpe, celle-ci en plans, comme illustré à la Figure 1-5 [KYAS95].

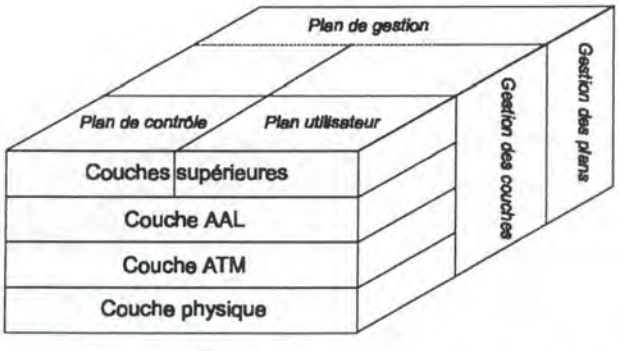


Figure 1-5 : découpe en plans et en couches du modèle ATM

Cette découpe en plans reflète les différents types de flux d'informations que le réseau doit supporter, introduisant la notion de flux d'information provenant de l'utilisateur, du contrôle et de la gestion (management) :

1. Plan utilisateur : le flux d'informations entre toutes les couches du modèle ATM se passe dans le plan utilisateur, qui s'occupe également de fonctions telles que le contrôle d'erreurs et le contrôle de flux.
2. Plan de contrôle : pour les transferts d'informations en mode connecté, il est nécessaire, tout comme dans le protocole X.25, d'établir une connexion, de la superviser et de la refermer lorsqu'elle n'est plus nécessaire. C'est le plan de contrôle qui s'occupe des fonction de signalisation.
3. Plan de gestion : ce plan est divisé en deux principales fonctionnalités : la gestion des couches et la gestion des plans. La gestion des couches traite de la méta-signalisation (*meta-signalling*), qui est en quelque sorte la signalisation pour la signalisation : il s'agit d'un protocole permettant la réservation

de canaux virtuels utilisés pour contrôler la signalisation. On se reportera au chapitre 2, section 2.3, pour un aperçu succinct sur le meta-signalling. Outre le meta-signalling, la gestion des couches gère également le trafic des cellules OAM (*Operation-Administration-Maintenance*) utilisées pour surveiller les performances du réseau. La gestion des plans est utilisée dans un but de coordination entre le plan de gestion et les deux autres plans (utilisateur et contrôle).

1.1.4 Notion de VPI, VCI et de commutation

1.1.4.a) VCI et VPI

Bien qu'ATM supporte des applications fonctionnant tant en mode connecté qu'en mode non connecté, ATM est un protocole fonctionnant toujours en mode connecté. Cela implique l'existence d'un chemin ou d'un canal entre les ressources ATM souhaitant être connectées entre elles (il peut s'agir de connexions point-à-point, point-à-multipoint ou multipoint-à-multipoint) qu'emprunteraient toutes les cellules devant circuler entre ces ressources. Tout se passe comme si l'on raccordait deux sites désireux de se connecter entre eux par un "tube", plusieurs tubes pouvant coexister sur le même support physique. Ce concept est équivalent à celui des circuits virtuels dans la technologie X.25. Dans la technologie ATM, chacun de ces tubes est unidirectionnel.

Il existe une hiérarchie dans ces "tubes" de transport de données, comme illustré à la Figure 1-6. Les canaux virtuels ou *virtual channels* (VC) se trouvent à la base de cette hiérarchie.

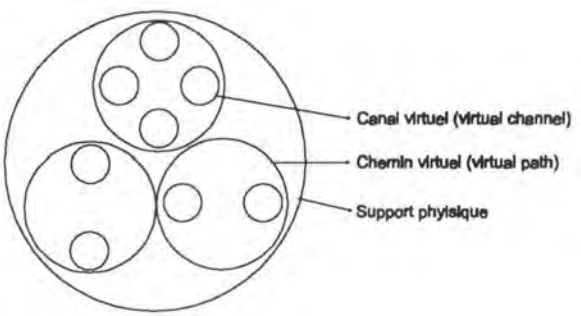


Figure 1-6 : hiérarchie dans les notions de chemins et canaux virtuels

Les quatre caractéristiques suivantes sont communes à tous les VC [KYAS95] :

1. Les paramètres de qualité de service (QoS) : à toute connexion de type VC sont assignés des paramètres de qualité de service spécifiant des caractéristiques telles que délai de cellule et taux de perte de cellules. Notons que si tout VC est caractérisé par une qualité de service, il n'est pas obligatoire que tous les VC contenus dans un même VP (le niveau hiérarchique directement supérieur) supportent la même qualité de service. On se référera à la section 1.1.5a pour une explication sur les notions de QoS.
2. Les connexions de type VC peuvent être permanentes ou établies via un protocole de signalisation.
3. La transmission en séquence des cellules est maintenue.
4. Des paramètres de trafic sont négociés entre l'utilisateur et le réseau au moment de l'établissement du VC. Les cellules passées au réseau par l'utilisateur sont surveillées afin de vérifier que les paramètres de trafic sont respectés. On se référera à la section 1.1.5b pour une explication sur la notion de paramètres de trafic.

On assigne à chaque VC un identifiant numérique unique appelé VCI pour *virtual channel identifier* qui a une signification locale uniquement.

Les chemins virtuels ou *virtual paths* (VP) se situent un niveau de hiérarchie au-dessus des VC. Un VP peut regrouper plusieurs VC. Nous expliquerons l'avantage de ce regroupement après avoir introduit la notion de commutation. Tout comme pour les VC, on assigne aux VP un identifiant numérique unique appelé *virtual path identifer* (VPI), ayant une signification locale uniquement.

Il existe deux types de VC ou VP : les permanents et les commutés (switched). Les VC et VP commutés sont des VC et VP établis par tout système de signalisation, ne nécessitant pas une configuration manuelle ou programmée à l'avance dans le nœud ou chez l'utilisateur. Les VC et VP établis par défaut au démarrage d'une machine, par l'intervention d'un opérateur ou par programmation préalable sont dit permanents.

Rappelons ici que les VCI et VPI ont une signification uniquement locale. Il en découle que les VC et VP sont également un concept uniquement local. L'ensemble des VC mis bout à bout entre deux utilisateurs connectés à travers un réseau ATM est appelé une connexion en canal virtuel (VCC : Virtual Channel Connection). De même, l'ensemble des VP mis bout à bout entre deux utilisateurs connectés à travers un réseau ATM est appelé une connexion en chemin virtuel (VPC : Virtual Path connection).

Nous verrons dans le chapitre 3 et le chapitre 4 deux protocoles de signalisation. Le premier, UNI 3.1 de l'ATM Forum, a pour but d'ouvrir une connexion entre deux équipements utilisateur ATM . Au terme du processus de signalisation un couple de valeur VPI/VCI est délivré aux utilisateurs, spécifiant ainsi le canal qui leur a été réservé. Notons cependant qu'il existe certaines valeurs VCI/VPI qui ne seront jamais allouées aux utilisateurs : il s'agit de valeurs réservées destinées au transfert d'informations entre certains protocoles tels UNI 3.1 et ILMI (se reporter au chapitre 3, section 3.2).

1.1.4.b) Notions de commutation

Nous avons dit à la section "*Format de cellules*" que l'adresse ATM n'était pas placée dans l'en-tête de la cellule, mais qu'en lieu et place on y mettait un label. Ce label est constitué d'un couple de valeur VPI / VCI utilisé comme identifiant unique d'une connexion, à signification uniquement locale. Remarquons que la connexion empruntée par une cellule est identifiée obligatoirement par le couple de valeur (VPI,VCI), ceci car deux VC appartenant à des VP différents peuvent porter le même identifiant VCI.

Il existe deux types de commutation dans les commutateurs ATM : la commutation de VP et la commutation de VC. Prenons l'exemple de la Figure 1-7 et concentrons-nous tout d'abord sur la notion de commutation de VP. La boîte grise représentant un commutateur ATM, deux VP entrent dans le commutateur (à gauche). Que ces VP arrivent par le même support physique ou non n'a que peu d'importance. Le VP entrant portant comme identifiant VPI = 4 est redirigé vers le VP de sortie portant l'identifiant VPI = 5. Tous les VC compris dans ce VP sont donc également redirigés, mais leurs identifiants ne sont pas changés. Le VP sortant portera le même identifiant jusqu'au prochain commutateur qui, s'il ne commute que les VP, changera à nouveau cet identifiant en sortie. Les commutateurs qui ne commutent que les VP sont appelés des *cross-connects*. L'avantage du cross-connect est d'avoir une vitesse de commutation très rapide, vu qu'il ne doit regarder et changer que le VPI dans chacune des cellules.

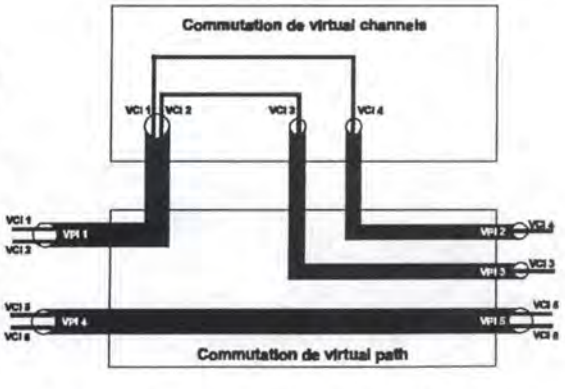


Figure 1-7 : commutation de VC et VP

Outre les cross-connects, il existe également des commutateurs qui commutent les VC. Ceci est représenté par le VPI = 1 dans la Figure 1-7. Dans ces commutateurs, les VC constituant un VP ne sont pas redirigés dans un même VP de sortie, mais dans deux ou plusieurs VP différents. Ces commutateurs demandent beaucoup plus de travail étant donné qu'il faut regarder et changer à la fois le VPI et le VCI. Ils sont aussi beaucoup plus onéreux que les cross-connects.

Quel est l'avantage d'avoir différencié la notion de VP et de VC ? Il y a deux avantages : l'efficience et le coût. Commençons tout d'abord par l'efficience. Supposons que deux sites universitaires soient interconnectés via un WAN ATM. Il est plus que probable qu'il y aura plus d'une connexion entre ces deux sites, chaque utilisateur dans une université souhaitant profiter du réseau et des services différents qu'il peut supporter. Il sera beaucoup plus efficient en terme de temps de commutation dans chacun des commutateurs par lesquels passeront ces connexions de regrouper tous ces différents trafics inter-universitaires dans un seul VP, au lieu d'avoir à commuter tous les VC. A cela s'ajoute également le fait que l'ouverture d'une connexion est nettement plus rapide si elle se fait à travers un VP déjà existant. Quand à l'avantage en terme de coût, il se situe plus au niveau de l'exploitation du réseau qu'à un niveau technique. De nombreux fournisseurs de services ATM loueront un ou plusieurs VP au gérant du réseau. Ces "ASP" (ATM Service Provider, par analogie aux ISP dans le domaine internet) sous-loueront des VC dans les VP loués aux gérants de réseaux ATM.

1.1.5 Contrat de trafic

Lorsqu'un utilisateur ATM désire ouvrir une connexion avec un autre utilisateur ou ressource ATM, il doit négocier, lors de cette phase d'ouverture, un contrat de trafic avec le réseau. Par ce contrat, il spécifie quel type de service il compte utiliser (cfr. les différentes classes de services proposées par les couches AAL). La négociation de ce contrat de trafic se fait par deux jeux principaux de paramètres : la qualité de service (QoS) et le descripteur de trafic de la connexion. Ce descripteur de trafic et la QoS sont spécifiés dans un message d'ouverture de connexion et examinés par chacun des commutateurs parcourus lors de la phase d'ouverture de cette connexion. L'entité chargée de vérifier si le trafic demandé peut être supporté par le commutateur s'appelle le *Call Admission Control* (CAC). Nous nous attarderons plus en détails sur le CAC dans le chapitre consacré à PNNI. Une fois le contrat accepté par le réseau, celui-ci s'engage à fournir le service demandé à l'utilisateur.

Après avoir présenté ces jeux de paramètres, nous développerons les procédures appliquées depuis la négociation du contrat jusqu'à l'utilisation de la connexion.

1.1.5.a) QoS

La qualité de service est utilisée afin de spécifier quel type de service est requis par l'utilisateur. Suivant qu'il s'agisse d'une application en temps réel nécessitant un flux variable de bits ou d'applications utilisant un flux constant de bits et peu sensibles à la synchronisation temporelle, ceci doit être clairement décrit lors de la négociation du contrat de trafic.

Il existe deux catégories de classes de service : la catégorie de QoS spécifiée et la catégorie de QoS non spécifiée.

i - QoS spécifiée

I. Généralités

La spécification I.362 de l'ITU-T a défini quatre classes de qualité de service utilisées pour chacune des classes de services offertes par les couches AAL, comme l'illustre le Tableau 1-2.

Rappelons qu'un VP peut contenir un ensemble de VC supportant des classes de services différentes. Dans ce cas, le VP doit garantir la QoS demandée la plus exigeante.

Si l'utilisateur souhaite ouvrir une connexion bidirectionnelle, il peut spécifier dans le message d'ouverture de connexion une classe de QoS pour chacune des directions. Ces classes ne doivent pas

nécessairement être les mêmes (exemple : liaison vidéo dans un sens et texte dans l'autre). Rappelons cependant que les VC et VP sont unidirectionnels.

Services offerts par couches AAL	Classes de QoS	Types d'application
Service de classe A (AAL type 1)	Classe QoS 1	Emulation de circuit, vidéo à débit constant
Service de classe B (AAL type 2)	Classe QoS 2	Audio et vidéo à débit binaire variable
Service de classe C (AAL type 3/4)	Classe QoS 3	Transfert de données orienté connexion
Service de classe D (AAL type 5)	Classe QoS 4	Transfert de données non-orienté connexion

Tableau 1-2 : classes de qualité de service

II . Paramétrisation de la QoS

L'association "classe de service A - classe de QoS 1" n'est pas définitive. En effet, cette association est totalement spécifique à l'implémentation : elle dépend du choix qu'aura fait l'administrateur d'un point d'accès au réseau. Celui-ci peut soit suivre la spécification I.362 de l'ITU-T pour l'association classe de service/QoS, soit définir sa propre association. Ceci implique que l'utilisateur a connaissance de l'association qui a été faite entre le terme QoS classe x et la définition même de cette qualité de service.

Au moment où est écrit ce mémoire, les spécifications des protocoles de signalisation aux interfaces UNI et NNI ne permettent pas de définir précisément une qualité de service en dehors des termes génériques "classe QoS 1, ..., 4, ou non spécifié". Cependant cette paramétrisation est introduite dans la version 4.0 du protocole UNI de l'ATM Forum en cours de spécification. La version définitive de UNI 4.0 devrait être votée par l'ATM Forum au mois de juin 1996.

Les paramètres qui permettent de caractériser une qualité de service font apparaître les notions de délai de transmission, de variation de celui-ci et de sensibilité l'erreur [TRAF4-96]. Le délai de transmission (*Cell Transfer Delay : CTD*) est négocié afin de définir le temps maximum séparant l'émission d'une cellule à un interface entre un utilisateur et le réseau (le côté émetteur du trafic) et la réception de cette cellule à un deuxième interface utilisateur / réseau (le côté récepteur du trafic). Le réseau devra faire tout son possible afin d'assurer que l'on ne dépasse pas ce délai. La variation du délai de transmission (*Cell Delay Variation : CDV*) est négociée afin de définir les seuils de variation minimum et maximum acceptables du délai de transmission. Pour des applications telles que la vidéoconférence ou toute application en temps réel ces limites seront très étroites et ne devront en aucun cas être dépassées. Le troisième paramètre, la sensibilité à l'erreur (*Cell Loss Ration : CLR*), est défini comme le quotient entre le nombre total de cellules perdues sur le nombre total de cellules transmises. Le réseau devra à nouveau faire tout son possible afin de rester en dessous du seuil CLR défini lors de la négociation du contrat de trafic.

ii - QoS non spécifiée

Lorsqu'un utilisateur ne connaît pas les types de service offert par le réseau qu'il va utiliser ou ne sait pas quel type d'association il y a entre la classe de service qu'il spécifierait et le service auquel elle correspondrait ou encore s'il ne prête aucune attention particulière au type de service que le réseau peut lui offrir, il a la possibilité de spécifier la classe de service non spécifié, dénommée "classe QoS 0". Dans ce cas, le réseau fournit le service qu'il peut à l'utilisateur. Cette classe particulière est utilisée généralement pour l'utilisation du service *Best Effort Capability*.

Avec le *Best Effort Capability*, l'utilisateur ne spécifie aucune classe de qualité de service particulière; il donne juste en paramètre le *Peak Cell Rate* (voir la section 1.1.5.b pour les notions de peak cell rate) pour les cellules de priorité CLP=1. Avec le Best Effort Capability, le réseau ne rejettera pas l'appel si le PCR spécifié est supérieur à la largeur de bande disponible.

1.1.5.b) Descripteur de trafic de la connexion

L'utilisateur doit définir au moment de l'établissement d'une connexion un ensemble de paramètres permettant de caractériser la classe de service demandée. Ces paramètres sont définis dans le *Source Traffic Descriptor*, défini sur les notions de paramètres de trafic et descripteur de trafic ATM.

- Paramètres de trafic : les paramètres de trafic sont une description quantitative d'un aspect particulier du trafic qui sera généré [UNI3.1-94]. Ces paramètres sont :

  1. *Peak Cell Rate* (PCR) : le PCR est défini comme étant l'inverse du temps minimum τ permis séparant l'arrivée de deux commandes provenant de la couche ATM demandant à la couche physique d'envoyer une cellule ATM [KYAS95].
  2. *Sustainable Cell Rate* (SCR) : le SCR spécifie une limite supérieure pour le taux moyen de transfert de cellules. C'est cette valeur que le réseau réserve par préférence au PCR.
  3. *Maximum Burst size* (MBS) : le MBS est le nombre de cellules maximum durant lequel l'utilisateur pourrait dépasser le SCR spécifié.

  Ces trois paramètres sont exprimés en nombre de cellules par seconde. L'intervalle de valeur admis pour ces trois paramètres est compris entre 0 et 16 777 215 cellules/sec.

  Exemple : au pire, mon application va générer un trafic de 100 Mbits/s pendant un temps maximum de 500 cellules et en général je ne dépasserai pas les 50 Mbits/s.

  Lors de la phase d'ouverture d'une connexion, ces paramètres peuvent être définis pour les cellules prioritaires (CLP=0) et non prioritaires (CLP=1). Dans ce cas, l'utilisateur spécifie toujours une classe de service (ou deux s'il désire une connexion bidirectionnelle), mais avec deux objectifs différents.

- Descripteur de trafic ATM (*ATM Traffic Descriptor*) : le descripteur de trafic ATM est une liste générique de paramètres de trafic utilisés pour définir les caractéristiques d'une connexion [UNI3.1-94]. L'utilisateur peut entre autres définir des paramètres de trafic pour les cellules prioritaires (CLP=0), non prioritaires (CLP=1), ainsi que pour une ou deux directions (*forward* et *backward*) selon qu'il désire une connexion unidirectionnelle ou bidirectionnelle. Le descripteur de trafic ATM est la liste exhaustive de tous les paramètres qui peuvent être utilisés afin de décrire les caractéristiques du trafic de toute connexion.

  Exemple : le descripteur de trafic ATM dans le protocole UNI 3.1 de l'ATM Forum (protocole faisant l'objet du chapitre 3) reprend l'ensemble des paramètres suivants : Forward PCR et Backward PCR pour les cellules CLP=0 et CLP=1, Forward SCR et Backward SCR pour les cellules CLP=0 et CLP=1, Forward MBS et Backward MBS pour les cellules CLP=0 et CLP=1, soit au total 12 paramètres.

- Descripteur de trafic source (*Source Traffic Descriptor*) : le descripteur de trafic source est un sous-ensemble de paramètres de trafic provenant du descripteur de trafic ATM, utilisé durant la phase de connexion afin de définir les caractéristiques intrinsèques du trafic de la connexion [UNI3.1-94]. Le descripteur de trafic source est propre à chaque connexion : il peut reprendre l'ensemble des paramètres du descripteur de trafic ATM mais peut également n'en reprendre qu'une partie, selon les caractéristiques de la connexion demandée. C'est cet ensemble de paramètres qui est utilisé lors de la négociation du contrat de trafic.

  Exemple : un utilisateur désirant avoir une connexion unidirectionnelle avec des cellules uniquement prioritaires pourrait avoir un descripteur de trafic source reprenant seulement les paramètres Forward PCR (CLP=1), Forward SCR (CLP=1) et Forward MBS (CLP=1).

1.1.5.c) Fonctions de gestion du trafic

Les fonctions utilisées afin de gérer le trafic, depuis sa négociation jusqu'à son utilisation, sont brièvement exposées ci-dessous :

1. *Fonction d'acceptation* : fonction exécutée par le CAC dans chacun des commutateurs parcourus lors de l'ouverture de connexion. Si l'un d'eux ne peut supporter le trafic demandé (spécifié dans le message de demande d'ouverture de connexion), soit l'appel est rejeté, soit le réseau va chercher un autre chemin susceptible de supporter le trafic demandé (on se reportera au chapitre sur PNNI pour plus de précisions sur cette procédure).
2. *Usage Parameter Control* (UPC) : les commutateurs prenant part à la connexion, une fois le contrat accepté, surveillent le trafic en provenance de l'utilisateur et vérifient si celui-ci respecte bien le contrat établi. Dans le cas où le contrat n'est pas respecté, cette fonction de maintien peut prendre certaines décisions, telles que éliminer purement et simplement les cellules "fautives", ou les marquer comme étant non-prioritaires. Etant marquées non-prioritaires, ces cellules sont susceptibles d'être supprimées par préférence à des cellules marquées prioritaires en cas d'engorgement du réseau.
3. *Fonction de comptabilité* : cette fonction est chargée de comptabiliser les frais découlant des paramètres de trafics négociés et devant être portés au débit du client (l'utilisateur).
4. *Traffic Shaping* : le *traffic shaping* est un ensemble de mécanismes résidant dans l'équipement de l'utilisateur permettant de modifier en cours de connexion les caractéristiques du trafic (diminuer le SCR par exemple). Notons que la QoS ne peut être modifiée en cours de connexion.

1.1.6 Adressage

Cette section reprend l'ensemble des formats d'adresses utilisés dans les réseaux ATM. Tous les formats d'adresses présentés dans cette section doivent être supportés obligatoirement par les réseaux ATM privés (maintenus par des organismes privés).

Les réseaux ATM publics (gérés par des organismes publics) ont le choix de supporter soit le format E.164, soit les formats d'adresses ATM privés présentés ci-dessous, soit les deux simultanément.

Une adresse ATM identifie de manière unique une ressource ATM. Le format d'une adresse est basé sur le format d'un *Network Service Access Point* (NSAP) OSI, comme spécifié dans les documents ISO 8348 et ITU-T X.213. Il existe trois formats différents d'adresses privées ATM, comme illustré par la Figure 1-8 [UNI3.1-94].

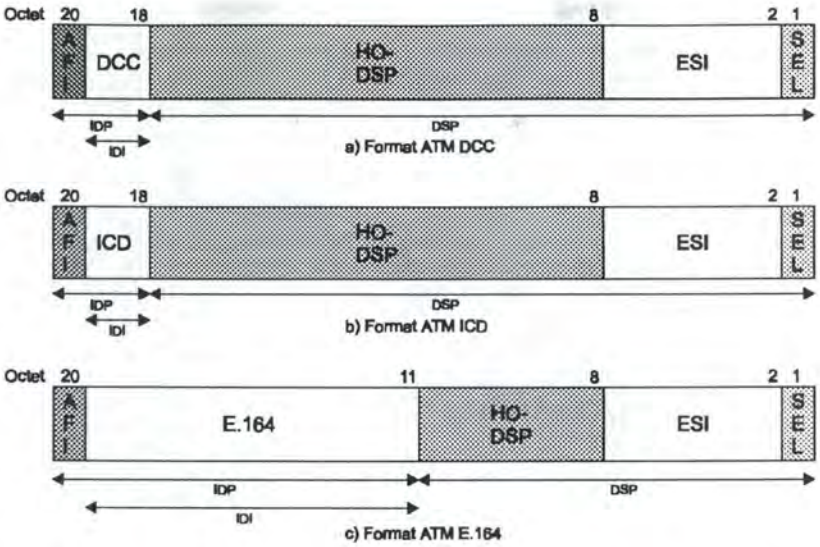


Figure 1-8 : format d'adresses ATM dans les réseaux privés

Reprenons les différents éléments composant ces adresses ATM [UNI3.1-94] :

1. *Initial Domain Part* (IDP) : l'IDP identifie de manière unique l'autorité administrative qui a la responsabilité d'allouer et d'assigner les valeurs du *Domain Specific Part* (DSP). L'IDP est constitué de deux champs : l'*Authority and Format Identifier* (AFI) et l'*Initial Domain Identifier* (IDI). L'AFI identifie l'autorité qui alloue le *Data Country Code* (DCC dans la Figure 1-8 a), l'*International Code Designator* (ICD dans la figure Figure 1-8 b), le numéro E.164 (dans la Figure 1-8 c), le format de l'IDI et la syntaxe du reste de l'adresse.
2. *Initial Domain Identifier* (IDI) : l'IDI spécifie le domaine d'adressage et l'autorité d'adressage pour les valeurs du *Domain Specific Part*. L'IDI est soit un code identifiant un pays (*Data Country Code*), soit un code identifiant un organisme international (*International Code Designator*), ou encore une adresse E.164.
3. *Data Country Code* : le DCD désigne le pays dans lequel l'adresse est enregistrée. Les valeurs du DCD sont spécifiées dans le document ISO 3166.
4. *International Code Designator* : l'ICD identifie un organisme international. Les valeurs et syntaxe d'encodage sont maintenus par le British Standards Institute.
5. E.164 : E.164 désigne les numéros pour réseaux de services à valeur ajoutée (RNIS ou ISDN : *Integrated Service Digital Network*), incluant donc les numéros de téléphone, pouvant compter jusqu'à 15 chiffres de longueur. Le champ E.164 dans une adresse ATM privée faisant 8 octets de longueur, on ajoutera en début de codage 4 bits ayant la valeur 0.
6. *Domain Specific Part* (DSP) : le DSP est constitué du *Domain Specific Part* de haut niveau (High Order-DSP ou HO-DSP) et de sa partie de bas niveau, constituée de l'identifiant de système terminal (ESI : *End System Identifier)* et du sélecteur.
7. *HO-DSP* : l'HO-DSP est défini par l'autorité identifiée par l'IDP et peut être utilisé de deux manières différentes : soit pour refléter la hiérarchie de l'organisme (dans les adresses ICD) ou des pays (dans les adresses DCC), soit pour faciliter le routage dans des réseaux ATM interconnectés.
    - Pour la hiérarchie : l'organisme international ou le pays identifié par l'IDP peut décider de subdiviser le champ HO-DSP en un ensemble de sous-champs qui permettront d'identifier dans ce pays/cet organisme le service ou la division qui aura attribué l'adresse.

- Pour le routage : le champ HO-DSP peut être subdivisé comme illustré à la Figure 1-9. Dans cette figure, le DFI identifie principalement la syntaxe du DSP. Les notions de domaine de routage (*routing domain*) et de zone dans ce domaine (*area domain*) permettent à l'organisme ou au pays identifié par l'IDP de morceler leurs réseaux privés en zones clairement identifiables. On consultera le document ISO 8348 ou le RFC 1237 pour plus de détails à ce sujet.

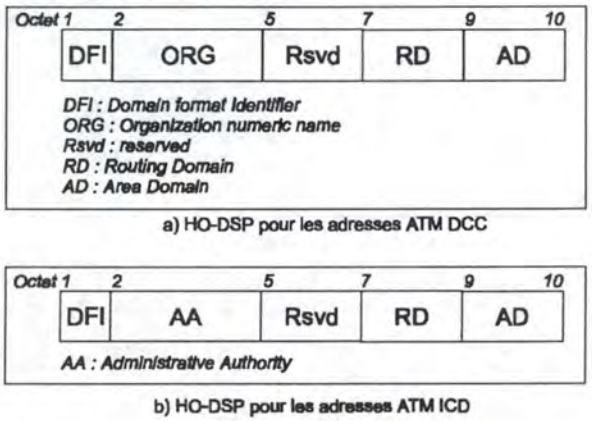


Figure 1-9 : découpe du champ HO-DSP pour les adresses DCC et ICD

8. *End System Identifier* : l'ESI identifie une ressource terminale ATM. Cette valeur doit être unique, en combinaison avec l'IDP et l'HO-DSP.
9. Sélecteur : non utilisé dans le routage ATM, mais peut être utilisé localement par les ressources ATM terminales afin d'identifier les entités du niveau couche application dans le modèle OSI qui doivent recevoir le trafic. Une manière d'identifier ces entités serait de placer dans le champ SEL le SAP correspondant à ces entités.

L'ensemble des différents composants d'une adresse donnés ci-dessus ne sont pas tous définis par l'ATM TE (Terminal Equipment). Celui-ci n'a de propre que l'ESI et le SEL. Tous les autres champs sont attribués par le point d'accès au réseau lors de la phase d'enregistrement des adresses (voir la section 3.2 du chapitre 3 pour plus d'informations à ce sujet). On parlera alors d'un "*Network Prefix*" pour tous les composants hormis ESI et SEL, eux-mêmes dénommés par le terme "*User Part*". Une adresse privée ATM est donc obtenue par la concaténation du network prefix et du user part.

La Figure 1-10 donne deux exemples d'adresses dans les réseaux ATM privés, codés ici en hexadécimal. La Figure 1-10 a) illustre une adresse DCC, b) illustre une adresse ATM ICD. Les valeurs des champs DCC et ICD sont correctes; les valeurs des champs ORG et AA de l'HO-DSP sont données à titre d'exemple et ne doivent pas être considérés comme correctes.

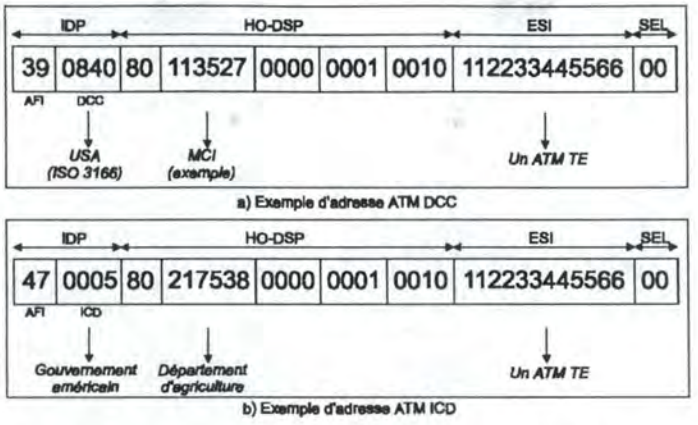


Figure 1-10 : exemples d'adresses ATM DCC et ICD

1.2 Propositions de lecture

Quelques ouvrages ou sites Internet sont proposés dans cette section au lecteur désireux d'approfondir ses connaissances sur le protocole ATM en général.

1.2.1 Ouvrages

- William STALLINGS, *ISDN and Broadband ISDN*, Second Edition, Macmillan Publishing Company, 1992
- Othmar KYAS, *ATM Networks*, Thomson International Publishing, 1995
- Uyless BLACK, *ATM : Foundation for broadband networks*, Prentice Hall Series In Advances Communications Technologies, 1995
- Martin de Prycker, *Asynchronous Transfer Mode - Solution for broadband ISDN*, Second Edition, Ellis Horwood Ltd., 1993

1.2.2 Internet

- *Norm Al Dude and Professor N. Erd on the subject of ATM* : http://www.datacomm-us.com/technow/scan06/scan06.html
- *The Cell Relay Retreat* : http://cell-relay.indiana.edu
- *ATM Home* : http://ganges.cs.tcd.ie:80/4ba2/atm/index.html
- *Asynchronous Transfer Mode (ATM)* : http://cne.gmu.edu/~sreddiva/Texttut.html
- *Michel Chirouze* : http://www.lirmm.fr/atm/

2. Architecture et protocoles utilisés pour la signalisation

Ce chapitre est consacré à la présentation des différentes couches constituant le modèle de signalisation dans les réseaux privés ATM. Sans nous attarder sur la couche de signalisation elle-même qui fera l'objet des deux prochains chapitres, nous développerons ici plus particulièrement les couches inférieures utilisées par cette couche de signalisation.

Une première section exposera les principes de la signalisation et les couches de protocoles utilisés dans un modèle général. Dans une seconde section, nous nous focaliserons sur la couche d'adaptation ATM spécialisée dans la signalisation : la couche *Signalling ATM Adaptation Layer* (SAAL). Une troisième section abordera le problème de l'allocation des VC utilisés pour le transfert des messages de signalisation et, par transition, des messages provenant de la couche SAAL. Une quatrième et dernière section tentera d'établir un rapport direct entre le modèle OSI et le modèle de signalisation ATM.

2.1 Présentation de l'architecture en couches pour la signalisation

Le chapitre 1 a présenté, entre autres, les couches du modèle ATM ainsi que leurs principales fonctions. Dans la Figure 1-5 de ce chapitre, trois couches se trouvant à la base de toute application utilisant la technologie ATM étaient exposées : couche physique, couche ATM et couche AAL. Une quatrième couche, dénommée dans cette figure "couches supérieures", y illustrait toute couche utilisant les services offerts par les trois premières. Dans le modèle de la signalisation sous ATM, cette couche supérieure est la couche de signalisation, utilisant une version particulière des couches AAL : la couche SAAL. Cette architecture est illustrée à la Figure 2-1.

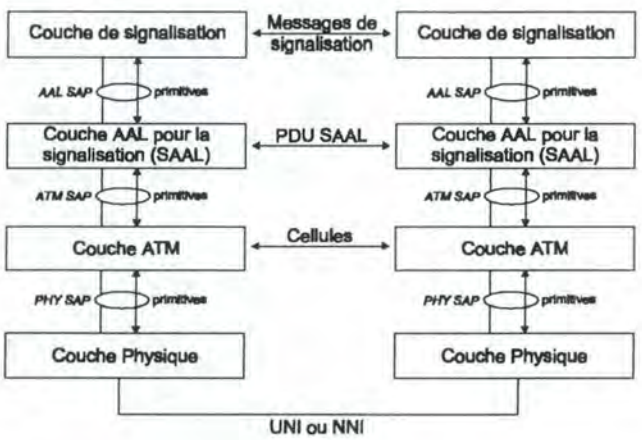


Figure 2-1 : architecture du modèle de signalisation

L'ensemble des couches impliquées dans le modèle de signalisation présentées dans cette figure feront l'objet d'une étude approfondie dans ce chapitre (SAAL et ATM) ainsi que dans les deux chapitres suivants (couches de signalisation).

Avant d'exposer les fonctionnalités et services offerts par les différentes couches impliquées dans le modèle de signalisation, il peut s'avérer nécessaire, pour le lecteur, de rappeler quelques termes de base repris dans cette figure :

- *Service Access Point* (SAP) : un SAP est l'interface se situant entre deux services offerts par deux couches adjacentes dans un modèle de communication décrit en couches. Dans la Figure 2-1 il y a un SAP entre chacune des couches impliquées dans le modèle de signalisation, à travers lesquels chaque couche offre des primitives de services à sa couche supérieure. Chaque couche supérieure à une autre s'enregistre auprès de celle-ci à travers un SAP. Ce SAP est propre à l'échange d'informations entre ces deux couches.
- *Primitive de service* : une primitive est un service offert par une couche à la couche qui lui est immédiatement supérieure (*upper layer*). Ce service peut être comparé à un appel de procédure. Une primitive est abstraite : elle n'est pas définie dans un langage de programmation bien particulier, mais représente une fonctionnalité qu'une couche doit offrir à sa couche supérieure. Dans la Figure 2-1, chacune des couches, à partir de la couche physique, offre des primitives de services à sa couche supérieure. Toute couche utilisant les services d'une couche qui lui est inférieure est le *Service User* de cette couche, cette dernière étant le *Service Provider* de la couche utilisant ses services.

Nous avons vu à la section 1.1.4a intitulée "VCI et VPI" qu'il existait deux types de VC : les VC permanents, établis manuellement par l'opérateur ou par configuration matérielle préalable et les VC commutés (*SVC : Switched Virtual Channels*), établis à l'aide d'un protocole de signalisation. Un protocole de signalisation est par définition un protocole utilisé pour l'ouverture, la maintenance et la fermeture d'une connexion - donc d'un VCC - entre deux ATM TE. Entre les deux entités de signalisation se trouvant dans chacun des deux TE, il existe un flux de messages : demande d'ouverture de connexion entre deux TE, acceptation ou refus de la connexion, demande de fermeture de la connexion, demande d'informations sur l'état de l'entité de signalisation distante, ... C'est suite à l'acceptation de la connexion par l'entité de signalisation distante que le VCC sera disponible au trafic d'information entre les deux TE.

Afin de pouvoir s'envoyer des messages de signalisation, les deux entités de signalisation distantes font appel aux couches inférieures de l'architecture. Celles-ci devront faire tout le nécessaire afin que ces messages arrivent bien à leur(s) destination(s). Nous verrons dans ce chapitre que le rôle de la couche SAAL, directement inférieure à la couche de signalisation, est d'offrir à cette dernière un transfert de données assuré sur un canal virtuel de signalisation qui sera ouvert uniquement pour le transfert de messages de signalisation. Le rôle de la couche ATM sera de convertir le flux d'information en provenance de la couche qui lui est directement supérieure (SAAL) en un flot de cellules. La couche physique, quant à elle, enverra ce flot de cellules à travers le réseau vers la destination.

Il y a deux protocoles de signalisation dans les réseaux privés : le protocole UNI (*User-to-Network Interface*) et le protocole PNNI (*Private Network-to-Network Interface*), tout deux développés par l'ATM Forum. Le protocole UNI 3.1 est destiné à l'échange de messages de signalisation entre un TE et son point d'accès au réseau (un commutateur). Ce commutateur relayera ensuite les messages de signalisation vers le point d'accès au réseau du TE distant par l'intermédiaire du protocole PNNI. Le point d'accès au réseau du TE distant utilisera enfin le protocole UNI 3.1 afin de faire parvenir les messages de signalisation au TE distant.

Nous avions introduit dans le premier chapitre une découpe en couches et en plans du modèle ATM. Pour rappel, le plan utilisateur concernait le transfert d'informations relatives à l'utilisateur, c'est-à-dire le trafic généré par cet utilisateur. Dans le modèle de la signalisation, nous nous trouvons dans un autre plan : le plan de contrôle. C'est ce plan qui, rappelons-nous, est responsable des opérations de signalisation.

2.2 Présentation des sous-couches utilisées

2.2.1 Couche SAAL

Comme exposé brièvement dans la section 2.1, le but de la couche SAAL est d'offrir à la couche de signalisation un transfert assuré des données en provenance de cette couche. En effet, la couche ATM est concernée uniquement par l'aspect "cellule" d'un transfert de données : elle n'assure aucun contrôle de flux ou d'erreurs, ne connaît que les données sous forme de cellules et ne peut en rien garantir la bonne émission/réception des données en provenance de l'utilisateur (dans ce cas, des données en provenance de la couche de signalisation). Ceci impliquerait alors que la couche de signalisation assure elle-même la sécurisation des transferts et la segmentation des messages à émettre sous forme de cellules, tâches sortant du cadre pur qui devrait être traité par un protocole de signalisation.

Il doit donc obligatoirement exister une couche intermédiaire entre la couche de signalisation et la couche ATM qui aura pour but d'offrir cette sécurisation ainsi que la segmentation des messages en provenance de la couche de signalisation. Cette couche particulière est la couche *Signaling ATM Adaptation Layer* (SAAL). La couche SAAL est une couche d'adaptation utilisée uniquement dans le cas de la signalisation. Elle offre à la couche de signalisation les canaux de transfert nécessaires au transport des messages de signalisation. Ces messages utilisent des canaux virtuels séparés du trafic d'informations en provenance d'un TE. Ce service est un service de transmission fiable. Par service de transmission fiable, on entend un transfert des données en mode connecté avec correction d'erreurs et contrôle de flux.

Avant de présenter l'architecture et les fonctionnalités de la couche SAAL, relevons une différence majeure entre la couche SAAL et les couches AAL utilisées pour le transfert d'informations utilisateur : alors que, comme l'illustrait la Figure 1-4, le transfert des informations relatives aux couches AAL dans le plan utilisateur ne concernait pas les commutateurs mais uniquement les entités AAL des TE distants, pour la couche SAAL - et donc pour le plan de contrôle - *tous* les TE et commutateurs impliqués dans une procédure de signalisation ont une couche SAAL et une couche de signalisation. Ceci implique que tous les commutateurs prenant part à une procédure de signalisation examinent chacun à leur tour les messages de signalisation et ne se contentent pas uniquement, comme c'est le cas dans le plan utilisateur, de servir d'aiguilleur de cellules. En effet, lors du transfert d'informations utilisateur un chemin a déjà été tracé entre les deux utilisateurs à travers le réseau (ce chemin a été tracé par l'intermédiaire des protocoles de signalisation). Chaque commutateur se trouvant sur ce chemin et recevant une cellule sait donc vers quel prochain commutateur il doit envoyer celle-ci. Par contre, lors d'une procédure de signalisation et plus particulièrement lors d'une ouverture de connexion à travers le réseau il faut qu'à chaque commutateur traversé on sache quelle est la destination désirée afin de décider vers quel commutateur il faut se diriger.

La couche SAAL est subdivisée en deux sous-couches comme illustré à la Figure 2-2 : la sous-couche *Service Specific Convergence Sublayer* (SSCS) dont le rôle est de fournir le service de transfert fiable des données et la sous-couche *Common Part* (CP) d'AAL5 dont le rôle est de segmenter les informations de la sous-couche SSCS (et par transition des informations en provenance de la couche de signalisation) en paquets de 48 octets pour la couche ATM qui elle-même les transformera en cellules. La sous-couche CP n'offre aucun mécanisme de transport fiable des données; ce rôle est, comme mentionné ci-dessus, pris en charge par la sous-couche SSCS.

Ces sous-couches ne communiquent pas entre elles par l'intermédiaire de primitives à travers un SAP mais via un ensemble de signaux. Ceux-ci seront exposés dans les prochaines sous-sections.

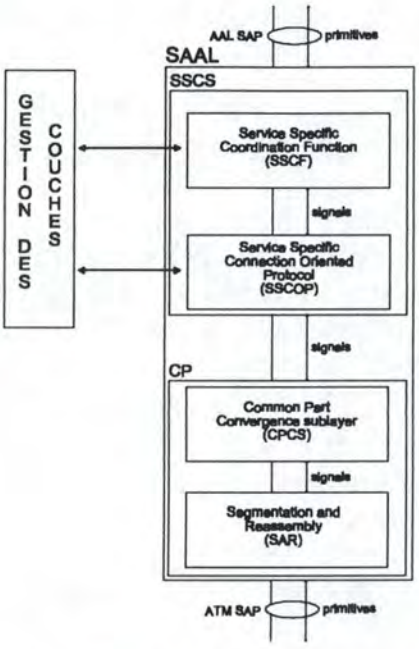


Figure 2-2 : découpe de la couche SAAL

La Figure 2-2 introduit également un autre concept : la gestion des couches. Nous avons vu que le transfert d'informations propres à l'utilisateur se faisait dans le plan utilisateur. Le plan de contrôle reprend quant à lui tous les protocoles impliqués dans la signalisation. Le dernier plan de la découpe en plan du modèle ATM est le plan de gestion. Il comprend entre autres une entité de gestion des couches. Dans les prochaines sections, nous ferons état de l'existence de transferts d'informations entre la sous-couche SSCF, ses composants et la gestion des couches (incluse, comme nous le savons, dans le plan de gestion). Le plan de gestion ne sera pas développé dans ce mémoire. Nous ferons cependant l'hypothèse de son existence. Son rôle principal est de fournir les fonctions de gestion et de maintenir les informations nécessaires au bon fonctionnement du réseau. Les protocoles utilisés dans le plan de gestion sont principalement SNMP (*Simple Network Management Protocol*) et ILMI (*Interim Local Management Interface*), décrits brièvement dans le chapitre 3.

A titre d'illustration, la Figure 2-3 [BLA95] reprend tous les différents protocoles qui peuvent coexister dans un TE.

Figure 2-3 : ensemble des protocoles résidant dans un TE

2.2.1.a) SSCS

Nous avons dit au début de ce chapitre qu'il est nécessaire pour la signalisation d'avoir une couche inférieure, la couche SAAL, qui offre un service de transfert fiable des données. Plus tard, nous avons dit que, dans cette couche SAAL, c'est la sous-couche SSCS qui offrait ce service de transfert fiable.

La sous-couche SSCS est elle-même fonctionnellement subdivisée en deux "blocs" ou modules : le *Service Specific Coordination Function* (SSCF) et le *Service Specific Connection Oriented Protocol* (SSCOP), comme illustré à la Figure 2-2. Parmi ces deux modules, le véritable "producteur" du service de transfert fiable offert par SSCS est le module SSCOP.

Différents protocoles de signalisation peuvent être utilisés au-dessus de la couche SAAL : UNI 3.1 et PNNI. Ces deux protocoles, bien que nécessitant les services de transmission fiable offerts par SSCOP, ne sont pas conçus pour être directement interfacés avec SSCOP. C'est le rôle du module SSCF que d'adapter les services offerts par SSCOP pour chacun de ces protocoles de signalisation. Il y a donc un module SSCF défini pour chacun des différents protocoles de signalisation.

i - SSCOP

Cette section a été rédigée sur base de [Q.2110-94].

Le module SSCOP est le "cœur" de la couche SAAL. Comme nous l'avons dit plus haut, le transfert assuré des données est offert par ce module. Afin d'assurer ce transfert fiable des données, SSCOP doit supporter un certain nombre de fonctions - exposées à la section suivante - dont les principales sont la correction d'erreurs et le contrôle de flux pour les informations en provenance de la couche de signalisation (par l'intermédiaire du module SSCF).

Un message de signalisation délivré par la couche supérieure est transféré sur un canal virtuel de signalisation dans des unités de données de longueur variable - les *Protocol Data Unit* (PDU) - propres au protocole qui les transmet. SSCOP est un protocole basé sur l'échange de PDU entre entités paires. Il s'agit d'un protocole de transfert de type "peer-to-peer". SSCOP génère ces PDU à partir des informations qui lui sont confiées par le module SSCF (i.e. le message de signalisation partiellement traité par SSCF - voir la section "SSCF" page 29). Soulignons qu'un message de signalisation peut se trouver être segmenté en plusieurs PDU SSCOP. Une autre fonction majeure de SSCOP sera alors d'assurer que l'on reçoit bien ces PDU en ordre, de manière à pouvoir recréer le message de signalisation original.

Une fois que toutes les informations nécessaires au contrôle de flux, d'erreurs et au respect de la séquence dans la transmission et réception des PDU ont été ajoutées aux informations reçues de la couche SSCF, SSCOP va utiliser les services offerts par la sous-couche CP afin de transformer les PDU SSCOP en champs de 48 octets qui seront placés par la suite dans des cellules par la couche ATM. La sous-couche CP est exposée à la section 2.2.1b.

Nous exposons dans la section suivante l'ensemble des fonctions qui doivent être supportées par le module SSCOP.

I. Fonctions de SSCOP

1. L'intégrité de séquence : le but de cette fonction est d'assurer un transfert des PDU SSCOP en séquence. Ceci est accompli en introduisant un numéro de séquence dans chacun des PDU à émettre. Un mécanisme de fenêtre coulissante est utilisé pour la transmission des PDU (technique comparable au protocole HDLC-LAP B).
2. La correction d'erreurs par retransmission sélective : la perte d'un PDU SSCOP est corrigée par une demande de retransmission du PDU manquant. Une erreur peut être détectée grâce au numéro de séquence introduit dans chacun des paquets : si après réception d'un PDU portant le numéro de séquence n, l'entité réceptrice remarque qu'elle n'a pas reçu le PDU de séquence n-1, elle en avertira l'entité émettrice qui retransmettra le PDU manquant. On consultera l'annexe B pour un exemple de détection et de correction d'erreurs.
3. Le contrôle de flux : l'entité SSCOP dans la station réceptrice peut contrôler le flux d'informations envoyé par l'entité SSCOP émettrice par des informations de contrôle de flux transportées dans les PDU SSCOP.

4. Le rapport d'erreur à la gestion des couches : toute erreur détectée dans le fonctionnement même du protocole SSCOP est signalée à la gestion des couches (gestion des couches dans le plan de gestion - voir la section 1.1.3 "Modèle en plans").
5. Le *Keep Alive* : lorsque deux entités SSCOP ne se sont plus envoyé de messages depuis une période de temps plus ou moins longue (paramétrable), les deux entités commencent à s'envoyer périodiquement des messages de service afin d'indiquer que la connexion est toujours nécessaire.
6. Le *Local Data Retrieval* : le but de cette fonction est de permettre à l'utilisateur de l'entité SSCOP de demander explicitement des PDU SSCOP que cette entité n'aurait pas encore acquittés ou supprimés. Cette fonction est utilisée par certains protocoles de signalisation dans les réseaux publics afin d'accroître la fiabilité de la livraison de messages : supposons par exemple que le VC utilisé pour le transfert de messages de signalisation subisse un problème technique majeur. Dans ce cas, le protocole de signalisation peut utiliser un autre VC, demander à SSCOP tous les messages qu'il n'a pu envoyer et les retransmettre sur le nouveau VC. Le protocole de signalisation SS7 (Q.704) utilise cette procédure. Les protocoles de signalisation UNI 3.1 et PNNI n'utilisent pas cette fonction.
7. Le contrôle de la connexion : cette fonction a pour rôle d'ouvrir, de fermer et de maintenir une connexion entre deux entités SSCOP. Elle permet également à l'utilisateur de cette entité de demander explicitement un transfert de données non garanti. Ceci se fait en plaçant directement la donnée utilisateur dans un message de service SSCOP (tel qu'une demande d'ouverture de connexion).
8. Le transfert de données utilisateur : cette fonction gère le transfert des données utilisateur entre deux entités SSCOP. Ce transfert peut se faire en mode garanti ou non garanti. Pour la signalisation, le transfert se fera toujours en mode garanti.
9. Le contrôle interne : cette fonction a pour but de détecter et de corriger des erreurs opérationnelles du protocole (tel qu'une erreur dans le header d'un PDU SSCOP).
10. L'échange d'informations de statuts entre les deux entités.

II. Signaux entre SSCOP et SSCF

Le module SSCF est, comme nous l'avons dit précédemment, utilisé pour interfacer la couche de signalisation avec le module SSCOP. Le module SSCF doit donc pouvoir effectuer des demandes à SSCOP, telles qu'une demande d'ouverture de connexion, de transfert garanti de données, de fermeture de connexion, etc.

SSCF va effectuer ces demandes par l'intermédiaire de signaux envoyés à SSCOP. De même, SSCOP signalera à SSCF la disponibilité d'une connexion, la fermeture de celle-ci ou la livraison de données en provenance de l'entité SSCOP distante par l'intermédiaire de signaux.

Nous exposons dans cette section les différents signaux échangés entre les modules SSCF et SSCOP. Ces signaux sont comparables à des primitives de services, bien que ce terme soit réservé à l'échange d'informations entre couches à travers un SAP (il n'y a pas de SAP entre les sous-couches et leurs modules). Ces signaux reposent sur le modèle request-indication-confirmation-response, illustré par la Figure 2-4.

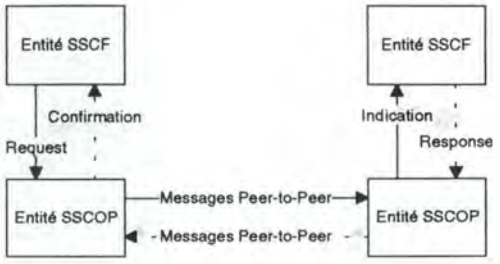


Figure 2-4 : modèle Request-Indication-Response-Confirmation pour les messages peer-to-peer

Le Tableau 2-1 reprend l'ensemble des signaux échangés entre une entité SSCOP et SSCF.

Signal	*Rôle*	*Paramètres en Request*	*Paramètres en Indication*	*Paramètres en Response*	*Paramètres en Confirmation*
AA-ESTABLISH	ouverture d'une connexion point-à-point pour un transfert de données garanti entre deux entités	SSCOP-UU	SSCOP-UU	SSCOP-UU	SSCOP-UU
AA-RELEASE	fermeture d'une connexion point-à-point entre deux entités	SSCOP-UU	SSCOP-UU source		/
AA-DATA	transfert point-à-point garanti de SSCOP SDU entre 2 entités	MU	MU SN		
AA-RESYNC	resynchronisation des buffers et des variables d'état de la connexion SSCOP	SSCOP-UU	SSCOP-UU	/	/
AA-RECOVER	procédure utilisée dans le recouvrement d'erreur		/	/	
AA-UNITDATA	transfert point-à-point non garanti entre entités SSCOP	MU	MU		
AA-RETRIEVE	primitive dédiée au local data retrieval	RN	MU		
AA-RETRIEVE COMPLETE	fin des SDU à remettre à l'utilisateur suite à un AA-RETRIEVE		/		
MAA-ERROR	rapport d'erreurs à la gestion des couches		code de l'erreur nombre de retransmissions		
MAA-UNITDATA	transfert de données point-à-point non garanti entre la couche SSCOP et la gestion des couches	MU	MU		

Tableau 2-1 : signaux et paramètres échangés entre SSCOP et SSCF

Les paramètres échangés à travers les différents signaux du Tableau 2-1 sont :

a) *SSCOP User-to-User information* (SSCOP-UU) : informations utilisateur de taille variable pouvant être transmises par l'intermédiaire d'un message de contrôle. Le transfert de cette information n'est pas garanti. La taille du SSCOP-UU peut être nulle. La signalisation n'utilise pas ce mode de transfert non garanti pour l'échange de messages entre entités de signalisation.

b) *Message Unit* (MU) : informations utilisateur de taille variable non nulle dont le transfert est garanti. Les informations en provenance de la couche de signalisation sont transportées dans ces MU.

c) *Retrieval Number* (RN) : ce numéro est associé à la procédure de local data retrieval : si l'utilisateur spécifie un numéro de type "unknown" (une valeur particulière), la couche SSCOP renvoie l'ensemble des PDU qu'elle n'a pas encore envoyés; un numéro de type "total" (une valeur particulière) renvoie l'ensemble des PDU se trouvant dans les buffers d'émission et la queue de transmission de SSCOP; une valeur N renvoie le PDU ayant le numéro de séquence N+1.

d) *Source* : la source indique à l'utilisateur SSCOP si la fermeture de connexion est due à la couche SSCOP elle-même ou à l'utilisateur de la couche SSCOP distante. S'il s'agit de la couche SSCOP, alors l'utilisateur ne doit pas tenir compte des informations éventuelles SSCOP-UU rendues par le signal d'indication.

e) *Sequence Number* (SN) : numéro de séquence du PDU reçu.

Chaque primitive génère des PDU différents. Ceux-ci sont définis dans la recommandation Q.2110 de l'ITU-T. On trouvera une description de ces PDU en annexe B.

III. SSCOP : une machine à états finis

SSCOP fonctionne sur base d'une machine à états finis. Chacun des états reflète les conditions générales de l'entité SSCOP suite à la séquence de signaux avec son utilisateur et l'échange de PDU avec l'entité à laquelle elle est connectée. Le Tableau 2-2 reprend la définition de l'ensemble de ses états.

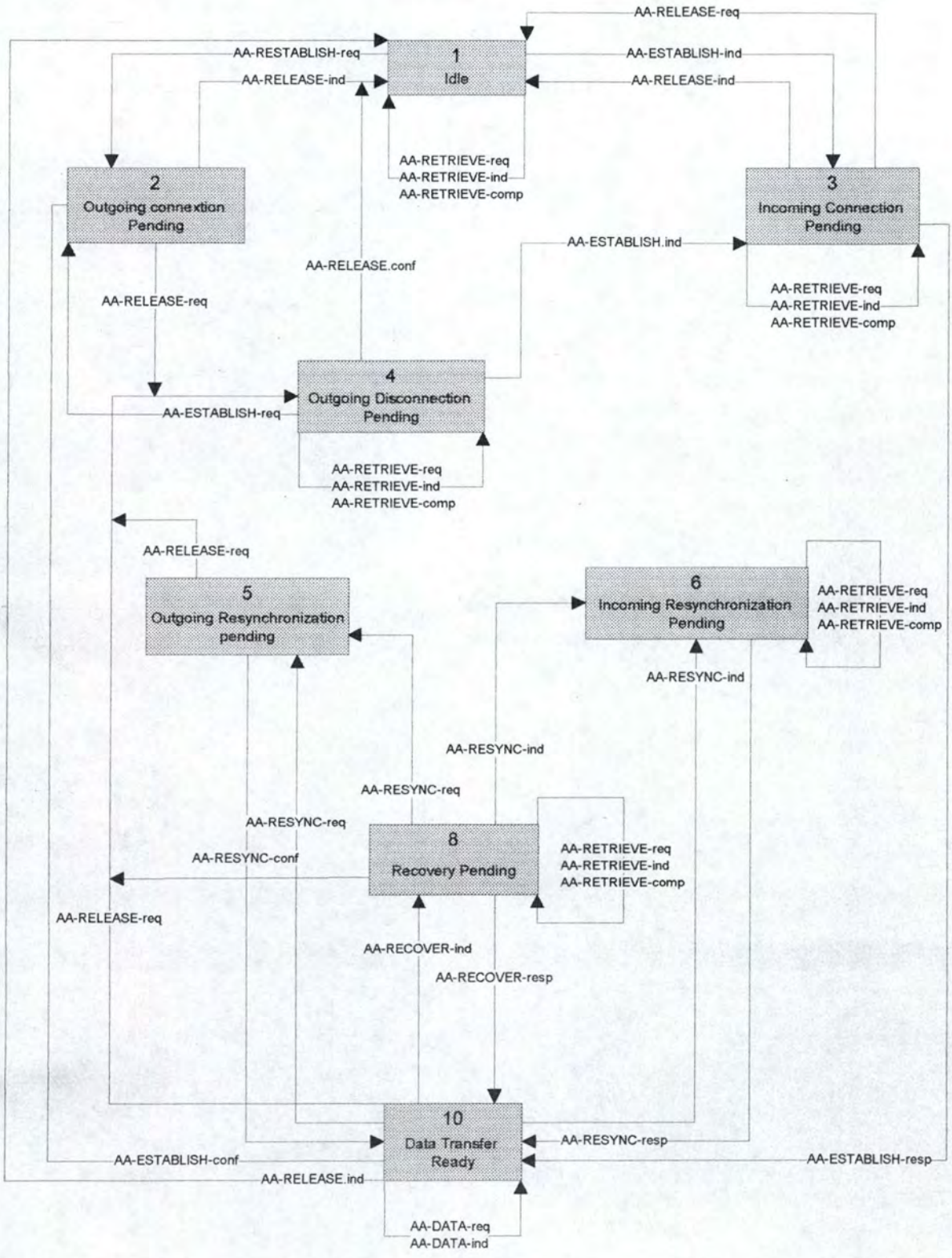
Etat 1 - Idle	Etat neutre au démarrage de SSCOP ou suite à une déconnexion
Etat 2 - Outgoing Connection Pending	L'entité SSCOP a effectué une demande de connexion avec l'entité paire
Etat 3 - Incoming Connection Pending	L'entité SSCOP a reçu une demande de connexion et attend la réponse de l'utilisateur (SSCF)
Etat 4 - Outgoing Disconnection Pending	L'entité SSCOP a effectué une demande de fermeture de la connexion; elle attend la confirmation que son entité paire a fermé la connexion et est retournée à l'état neutre
Etat 5 - Outgoing Resynchronization Pending	L'entité SSCOP a effectué une demande de resynchronisation avec l'entité paire
Etat 6 - Incoming Resynchronization Pending	L'entité SSCOP a reçu une demande de resynchronisation et attend la réponse de son utilisateur
Etat 7 - Outgoing Recovery Pending	L'entité SSCOP a demandé à son entité paire une récupération d'erreur
Etat 8 - Recovery Response Pending	L'entité SSCOP a corrigé l'erreur et l'a signalé à son utilisateur
Etat 9 - Incoming Recovery Pending	L'entité SSCOP a reçu le signalement qu'une erreur s'est produite par son entité paire; elle en a notifié son utilisateur et attend sa réponse
Etat 10 - Data Transfer Ready	L'entité SSCOP est prête à transférer des données en mode garanti

Tableau 2-2 : description des états associés à la FSM de SSCOP

La Figure 2-5 [Q.2110-94] reprend les transitions majeures suite à l'échange de signaux entre SSCOP et SSCF. Notons que dans cette figure, l'état 8 *Recovery Pending* représenté couvre l'état 8 *Recovery Response Pending* ainsi que l'état 9 *Incoming Recovery Pending*. Ceci est dû au fait que l'état réellement en cours n'est pas visible à l'interface SSCF/SSCOP [Q.2110-94]. L'état 7 *Outgoing Recovery Pending* n'est jamais visible à l'interface SSCF/SSCOP [Q.2110-94] et n'est donc pas représenté à la Figure 2-5.

ERRATA

- Figure 2-5, page 29 : schéma incomplet



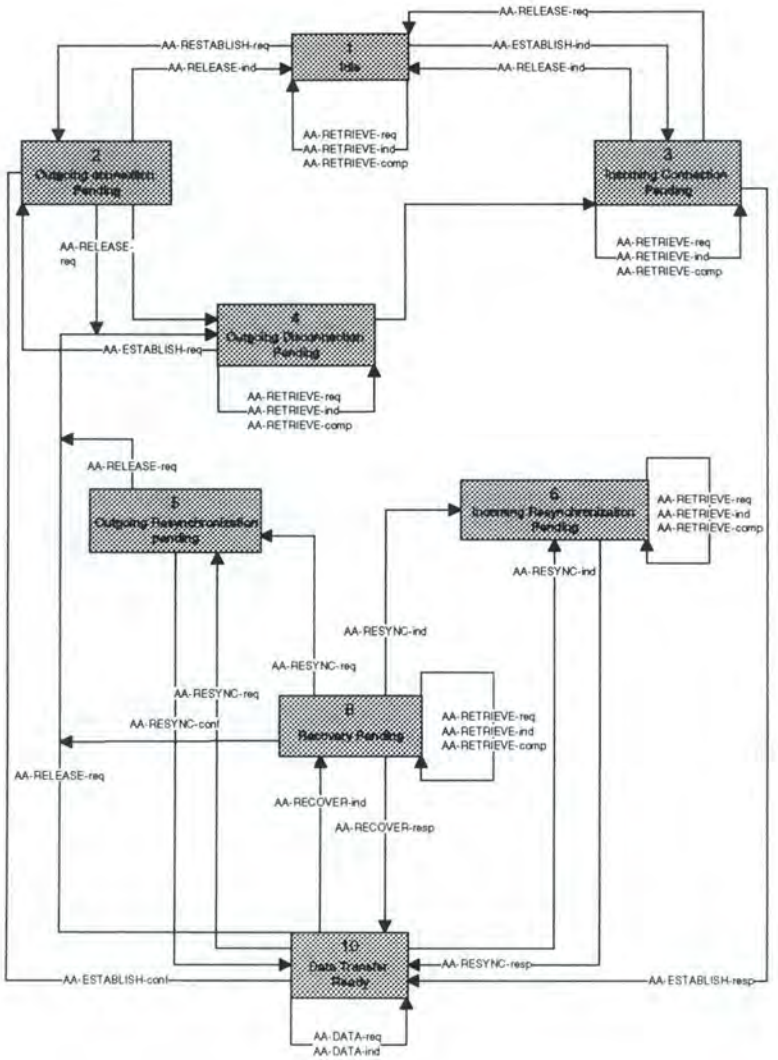


Figure 2-5 : transitions des états SSCOP à l'interface avec SSCF

ii - SSCF

SSCF est l'entité chargée d'assurer le dialogue entre SSCOP et l'utilisateur de service de SAAL : l'entité de signalisation. Utilisant les services offerts par SSCOP, SSCF offre à la couche directement supérieure (la couche de signalisation) un canal de signalisation sûr pour le transfert des messages de signalisation. SSCF transforme les requêtes de l'entité de signalisation en signaux directement compréhensibles par la couche SSCOP. Il s'agit donc principalement d'un protocole d'interfaçage entre SSCOP et toute couche de signalisation. La section IV page 33 illustrera l'utilisation de SSCOP par SSCF dans une procédure de demande d'établissement d'une connexion. SSCF n'offre aucun autre mécanisme de contrôle de séquence, de contrôle d'erreurs et de contrôle de flux que celui de SSCOP.

SSCF est également en relation avec la gestion des couches afin de veiller à une utilisation correcte des canaux virtuels de signalisation. Nous verrons entre autres dans la section "Etapes avant transfert" qu'avant de mettre un canal virtuel de signalisation à la disposition du protocole de signalisation, SSCF enclenche tout d'abord une période de test afin de vérifier la qualité du canal. La gestion des couches communique également avec SSCF pour demander la fermeture d'un canal virtuel qui n'aurait pas satisfait à la période de test.

Contrairement à SSCOP, il n'y a pas un seul type de SSCF mais un SSCF défini pour les différents types de besoin des couches supérieures.

Dans le cas de la signalisation dans les réseaux ATM, il existe un SSCF pour le protocole de signalisation *Private Network-to-Network Interface* (PNNI) [Q.2140-94] et un SSCF pour le protocole de signalisation *User-to-Network Interface* (UNI) [Q.2130]. Des SSCF peuvent également être spécifiés pour des applications autres que la signalisation tel FR-SSCF (I.365.1) qui définit une couche SSCF à utiliser dans les cas d'interworking Frame Relay - ATM.

Le protocole SSCF présenté dans ce chapitre correspond à la recommandation Q.2140 "*B-ISDN Signalling ATM Adaptation Layer - Service Specific Coordination Function for support of signalling at the network interface (NNI)*" de l'ITU-T.

1. Etapes avant transfert

Nous avons dit à la section précédente que SSCF était en contact avec la gestion des couches pour effectuer une période de test sur la qualité du canal virtuel de signalisation ouvert suite à une demande d'ouverture de ce canal par le protocole de signalisation. Pour ce faire, SSCF passe par une série d'états d'alignement en ouvrant la connexion pour le protocole de signalisation. Cette période d'alignement utilise des fonctionnalités de contrôle d'erreurs propre à la gestion des couches. Les fonctionnalités de contrôle d'erreurs de la gestion des couches ne seront pas exposées dans ce mémoire. Nous ferons cependant l'hypothèse de l'existence de ces fonctionnalités. Le lecteur désireux d'approfondir ce sujet peut se référer à la recommandation Q.2144 "*B-ISDN ATM Adaptation Layer - SSCS Layer Management at the UNI*" de l'ITU-T.

Les différents états d'alignement par lesquels passe l'entité SSCF sont :

1. *Out of service* : aucune connexion n'existe; c'est l'état nul.
2. *Alignment* : avec la phase d'alignement commence la période de test du lien. Cette phase est initiée par une demande de connexion de la part de l'utilisateur.
3. *Test* : une fois que SSCF a eu notification de l'ouverture du lien, il demande à la gestion des couches de commencer la période de test. Durant la période de test (dont la durée est définie par un timer réservé à cet effet), SSCF va envoyer une série de PDU à son entité paire. L'entité de gestion des couches distante en vérifiera la bonne réception, puis les effacera.
4. *Alignment Ready*: une fois le nombre approprié de PDU envoyé durant la période de test, SSCF notifie à l'entité de gestion des couches et à l'entité SSCF paire (par l'envoi d'un PDU particulier) que la période de test est terminée. Dès la réception d'un PDU de confirmation provenant de l'entité SSCF paire, SSCF signale à l'entité de gestion des couches et à l'entité de signalisation que le lien est établi et entre dans l'état *In Service*. Si d'autre part SSCF reçoit de son entité paire un PDU de fin de test alors que la période de test n'est pas encore terminée, SSCF entre directement dans l'état *In Service* et le notifie à l'entité de gestion des couches et à son utilisateur.
5. *In Service* : le lien est disponible à l'utilisation.

Le transfert entre ces différents états est illustré à la Figure 2-6 [Q.2140-94]. Les primitives figurant dans ce schéma et entraînant la transition d'un état vers un autre sont exposées à la section suivante.

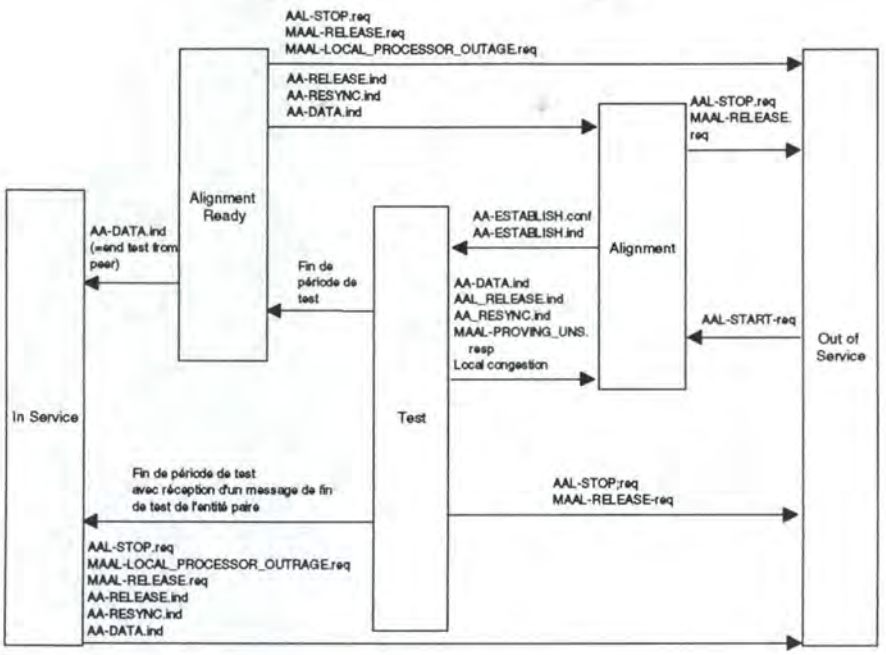


Figure 2-6 : transition entre états dans la préparation d'un lien par SSCF

II . Primitives offertes par SSCF

Le Tableau 2-3 reprend l'ensemble des primitives de service offertes par SSCF à la couche supérieure. Il s'agit bien ici de primitives et non de signaux, puisque l'on se trouve à l'interface entre deux couches : la couche SAAL et la couche de signalisation. On trouvera un exemple d'utilisation de ces primitives pour l'ouverture d'une connexion suite à une demande émanant de l'entité de signalisation à la section IV page 33.

Outre le paramètre *Message Unit* (MU) que l'on a déjà rencontré dans SSCOP, trois nouveaux paramètres sont utilisés dans SSCF :

1. Paramètre de congestion : paramètres utilisés dans le mécanisme de contrôle de flux (cfr. note 1 page 32). Ils sont définis suivant la recommandation Q.704 de l'ITU-T.
2. *Backward Sequence Number to be Transmitted* (BSNT) : paramètre utilisé pour le Local Data Retrieve. SSCF permet donc à l'entité de signalisation d'utiliser la fonction Local Data Retrieve de SSCOP. Lorsque l'utilisateur de SSCF génère un AAL-RETRIEVE_BSNT.request, SSCF s'assure tout d'abord qu'il a traité tous les AA-DATA.ind provenant de SSCOP. SSCF génère alors un AAL-BSNT.confirm en rendant le paramètre BSNT qui correspond au numéro de séquence du PDU reçu dans le dernier AA-DATA.indication.
3. *Forward Sequence Number of last message signal unit accepted by peer* (FSNC) : paramètre utilisé pour le Local Data Retrieve. Lorsque l'utilisateur de SSCF génère un AAL-RETRIEVAL_REQUEST_AND_FSNC.request, SSCF génère un AA-RETRIEVE.request à SSCOP (voir "I Fonctions de SSCOP" page 25 et "II Signaux entre SSCOP et SSCF" page 26) en utilisant FSNC comme Retrieval Number. SSCF retourne alors à l'utilisateur les PDU qu'SSCOP lui aura rendu par l'intermédiaire du signal AA-RETRIEVE.indication en utilisant la primitive AAL-RETRIEVED_MESSAGES.indication. Lorsqu'il n'y a plus de messages à retourner à l'utilisateur, SSCF génère un AAL-RETRIEVAL_COMPLETE.indication.

Primitives	*Request*	*Indication*	*Response*	*Confirmation*	*Rôle*
AAL-MESSAGE-FOR-TRANSMISSION	MU				Données (MU) à émettre
AAL-RECEIVED-MESSAGE		MU			Réception de données utilisateur
AAL-STOP					Demande de fermeture de connexion
AAL-START					Demande d'établissement de connexion
AAL-IN-SERVICE					Lien disponible
AAL-OUT-OF-SERVICE					Lien non disponible
AAL-LINK-CONGESTED		paramètre de congestion			Indication de congestion[1]
AAL-LINK-CONGESTION-CEASED					Fin des problèmes de congestion
AAL-RETRIEVE-BSNT					Demande de récupération du BSNT
AAL-BSNT				BSNT	Livraison de la valeur BSNT
AAL-RETRIEVAL_REQUEST_AND_FSNC	FSNC				Demande des messages non encore acquittés.
AAL-RETRIEVED_MESSAGES		MU			Livraison des messages non encore acquittés
AAL-RETRIEVAL_COMPLETE					Fin de la récupération des messages non acquittés
AAL-BSNT_NOT_RETRIEVABLE					Il n'y a pas de BSNT correspondant à retourner

Tableau 2-3 : primitives de services offertes par SSCF

III. Interfaçage avec la gestion des couches

Comme nous l'avons dit à la section "*Etapes avant transfert*", SSCF interagit avec la gestion des couches pour la période d'alignement avant transfert ainsi que lors de la fermeture du canal virtuel de signalisation si celui-ci n'a pas satisfait les tests de transfert durant la période d'alignement. Ceci se fait par l'intermédiaire de signaux, exposés dans le Tableau 2-4. La section IV illustre par un exemple l'utilisation de ces signaux.

[1] AAL-LINK-CONGESTED et AAL-LINK-CONGESTION-CEASED sont deux primitives intervenant dans le contrôle de flux offert par la couche SAAL. SAAL veille à ne pas émettre un ou plusieurs PDU qui pourraient enfreindre le contrat de trafic qui a été négocié. SAAL retarde alors l'émission du (des) PDU et en avertit l'utilisateur par la primitive AAL-LINK-CONGESTED.indication

Signaux	*Request*	*Indication*	*Response*	*Direction*	*Usage*
MAAL-PROVING				SSCF à GC	Demande de démarrage de la période de test
MAAL-STOP_PROVING				SSCF à GC	Fin de la période de test
MAAL-PROVING_UNSUCCESSFUL				GC à SSCF	Période de test non réussie
MAAL-FORCE_PROVING				GC à SSCF	Demande d'une période de test par la gestion des couches
MAAL-FORCE_EMERGENCY				GC à SSCF	Demande expresse de ne pas effectuer la période de test
MAAL-CLEAR_FORCE_MODES				GC à SSCF	Indication que la gestion des couches est indifférente au déroulement d'une période de test
MAAL-RELEASE				GC à SSCF	Demande de fermeture de la connexion
MAAL-REPORT		(frontière basse, frontière haute, raison pour condition exceptionnelle)		SSCF à GC	Indication à la gestion des couches de certains événements

Tableau 2-4 : signaux échangés entre la gestion des couches (GC) et SSCF

Les notions de frontière basse, frontière haute et raison en cas de condition exceptionnelle dans le signal MAAL-REPORT se réfèrent à des événements se déroulant soit à la frontière basse ou haute de SSCF : passage en état In Service, Out Of Order, fermeture de la connexion par SSCOP, par l'utilisateur ou par l'entité distante, phase d'alignement, détection de congestion, etc. Elles permettent à la gestion des couches d'avoir une vue bien définie de l'état dans lequel se trouve SSCF.

IV. Exemple

La Figure 2-7 illustre une demande d'ouverture de connexion entre deux entités SSCF suite à une demande de l'entité de signalisation.

Dans cette figure, l'entité de signalisation de gauche demande à l'entité SSCF d'établir une connexion avec l'entité de signalisation de droite. La figure illustre l'utilisation par l'entité SSCF des services de SSCOP, ainsi que les interactions entre SSCF et la gestion des couches. Les deux entités SSCOP conversent par l'intermédiaire de PDU SSCOP, décrits en annexe B.

Une fois la période d'alignement commencée, l'entité SSCF de gauche envoie, comme expliqué à la section I "Etapes avant transfert" de la page 30, une série de PDU afin de vérifier la qualité du lien. Ce transfert de PDU se fait soit entre un TE et son point d'accès au réseau soit entre deux commutateurs, mais toujours entre deux entités SSCF paires. Rappelons-nous en effet que dans le plan de contrôle, chacun des commutateurs ou TE doit traiter les informations transmises jusqu'à la couche SAAL et la couche de signalisation, à l'instar du transfert de données dans le plan utilisateur.

Le lecteur remarquera que les entités SSCOP envoient toujours un POLL PDU après l'émission d'un SD PDU. Ceci n'est absolument pas obligatoire en cas de transfert "normal" de données, mais permet de vérifier durant la période d'alignement qu'après chaque envoi d'un SD PDU celui-ci a été bien reçu.

Remarquons également l'utilisation du signal MAAL-REPORT.ind. Celui-ci n'est utilisé que dans le cas où un changement a pu être observé dans l'état de SSCF : immédiatement suite à une demande d'ouverture de connexion (SSCF passe de l'état *out of service* à l'état d'alignement) ou suite à la mise à disposition du lien (SSCF passe à l'état *in service*).

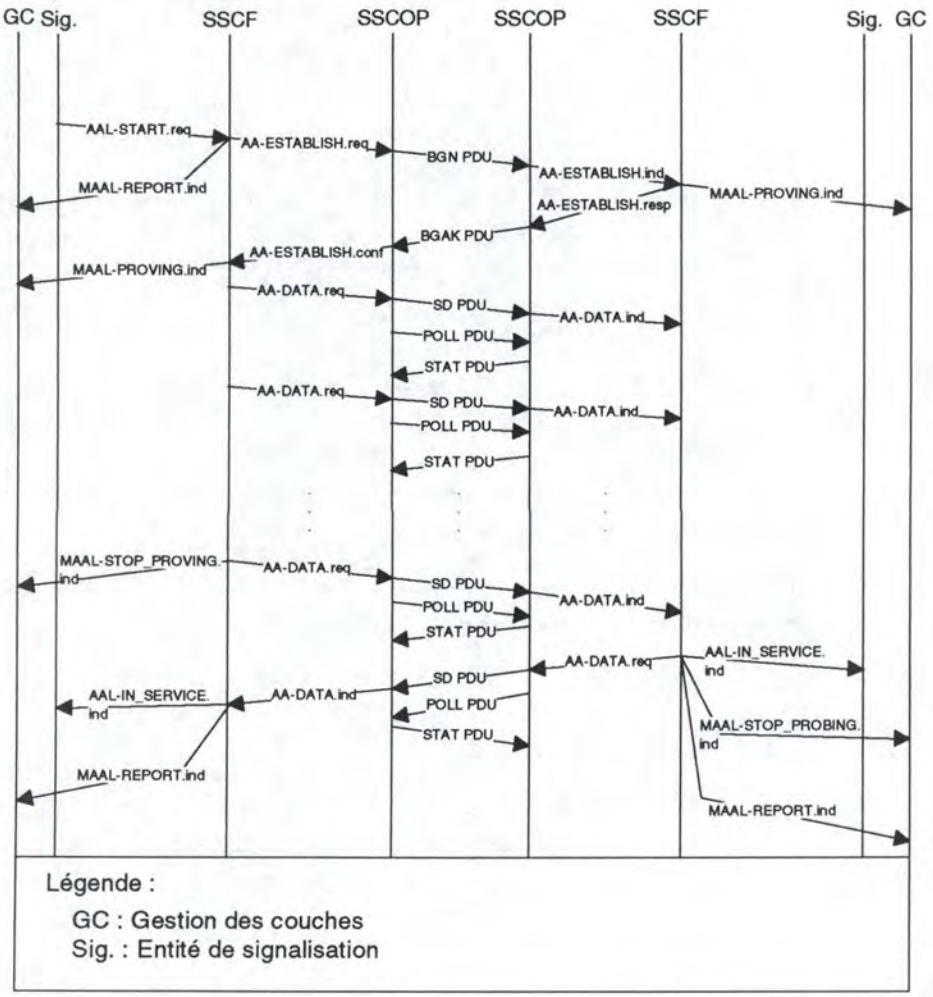


Figure 2-7 : exemple d'ouverture de connexion entre entités de signalisation

2.2.1.b) CP

Nous avons dit à la section 2.1 que l'unique rôle de la sous-couche CP est de convertir les PDU SSCOP en paquets de 48 octets, délivrés ensuite à la couche ATM qui les transformera en cellules.

SSCOP fournit ces PDU à la sous-couche CP par l'intermédiaire du signal *CPCS-UNITDATA.invoke (Interface Data)*, où *Interface Data* (ID) est le paramètre contenant le PDU à transmettre. La sous-couche CP utilisée est la sous-couche CP d'AAL5. AAL5 a été choisi pour sa simplicité de mise en œuvre (AAL5 est une version simplifiée de AAL 3/4).

La taille des PDU pouvant être confiés à la sous-couche CP peut varier entre 1 et 65535 octets.

Dans la sous-couche CP, un premier module, le *Common Part Convergence Sublayer* (CPCS) ajoute au PDU SSCOP un champ de padding afin d'arriver à un paquet dont la taille globale est un multiple de 48 octets. Ce champ de padding doit tenir compte d'un deuxième champ ajouté au PDU SSCOP, le *trailer*, dont la taille est 8 octets (2 octets réservés, 2 octets enregistrant la taille du PDU SSCOP et 4 octets de CRC sur l'ensemble du paquet).

Ce paquet est ensuite confié au module de segmentation et réassemblage (SAR) qui le segmentera directement en paquets de 48 octets chacun. Ces paquets de 48 octets seront directement insérés dans une cellule par la couche ATM.

De même, la sous-couche CP délivre à SSCOP un PDU reconstitué à partir de l'ensemble des paquets de 48 octets qui lui auront été confiés par la couche ATM. Le SAR ayant reconstitué le PDU, CPCS vérifie qu'il y a bien correspondance entre le champ "longueur" du trailer et la longueur totale du PDU; il vérifie également le CRC. Si ces deux tests sont positifs, CPCS retourne le PDU à SSCOP par l'intermédiaire du signal *CPCS-UNITDATA.signal (Interface Data)*. Remarquons que le PDU CPCS transporte la taille du paquet uniquement dans le trailer. De ce fait, l'entité CP réceptrice ne peut savoir, en recevant les premières cellules, quelle sera la taille totale du paquet. Elle devra alors toujours allouer un buffer de 65535 octets à la réception. Si le paquet n'est pas complètement reçu après avoir rempli ce buffer, c'est qu'il y a une erreur.

2.3 Allocation des canaux de signalisation

Lorsqu'un utilisateur communique avec un autre utilisateur ou ressource ATM à travers le réseau, il le fait en utilisant des canaux virtuels établis lors de la phase de signalisation. Lors de la phase même de signalisation, des messages sont échangés au travers du réseau (ou des réseaux) entre les deux utilisateurs ou ressources ATM désirant se connecter afin d'ouvrir et réserver ces canaux virtuels. Or les messages de signalisation doivent également emprunter des canaux virtuels afin d'être acheminés à destination. Il y a deux possibilités afin d'aborder ce problème :

1. il est nécessaire d'utiliser un autre protocole de signalisation, de plus bas niveau, qui aura pour but d'allouer un couple de valeurs VPI/VCI au protocole de signalisation "proprement dit" (par exemple UNI ou PNNI). Ce protocole existe, c'est le "meta-signaling. Il a été spécifié par l'ITU et porte la référence Q.2120. Une entité de méta-signalisation, appelée MSPE (*Meta-signaling Protocol Entity*), se situe au niveau de la gestion des couches de la couche ATM. Un MSPE utilise les services de la couche ATM pour le transport des messages de méta-signalisation et communique avec la gestion des plans pour la réservation des VC de signalisation.
2. il existe des couples de valeurs VPI/VCI réservés intentionnellement pour l'utilisation de protocoles spécifiques. Ainsi, il n'est pas nécessaire d'utiliser un protocole de méta-signalisation.

Les protocoles UNI et PNNI n'utilisent pas la méta-signalisation. En effet, des canaux virtuels sont réservés de manière permanente et exclusive pour ces deux protocoles. Le Tableau 2-5 reprend les couples de valeurs VPI/VCI réservés pour les protocoles abordés dans ce mémoire.

Notons qu'un couple VPI/VCI est alloué d'office pour la méta-signalisation. Si tel n'était pas le cas, cela entraînerait la nécessité de l'existence d'un protocole de "méta-méta-signalisation", problème comparable à celui de la poule et de l'œuf.

	VPI	*VCI*
UNI	0	5
Signalisation PNNI	0	5
Routage PNNI	0	18
ILMI	0	16
Meta-signaling	0	1

Tableau 2-5 : couples de valeurs VPI/VCI réservés

2.4 Comparaison avec le modèle OSI

Comme nous l'avons dit dans la section 1.1.2 du chapitre 1, il serait erroné d'associer directement la couche 1 du modèle OSI avec la couche physique ATM, la couche 2 OSI avec la couche ATM, la couche 3 OSI avec la couche SAAL (AAL), etc. Cette section tentera d'établir un parallèle entre la découpe en couches du modèle OSI et celle du modèle ATM, la fonctionnalité des différentes couches intervenant dans le modèle de signalisation étant maintenant connue.

Exposons tout d'abord les fonctionnalités des trois premières couches intervenant dans le modèle OSI, telles que définies par l'ISO [STA92] :

1. *Couche physique* : la couche physique traite de la transmission d'un flot non structuré de bits sur un support physique. Ceci implique la gestion de paramètres tels que le voltage du signal, la durée de l'émission de chaque bit, ... La couche physique traite des caractéristiques mécaniques, techniques et procédurales afin d'établir, maintenir et désactiver le lien physique.
2. *Couche logique* : la couche logique assure un transfert fiable des données à travers le lien physique. Elle envoie des blocs de données (trames) en assurant la synchronisation nécessaire, le traitement des erreurs et le contrôle de flux.
3. *Couche réseau* : la couche réseau permet aux couches supérieures de ne pas se soucier des technologies de transmission de données et des techniques de commutation utilisées pour interconnecter les systèmes. Elle est responsable de l'établissement, du maintien et de la fermeture de connexion.

Au vu de ces trois définitions, nous pouvons établir la comparaison suivante entre modèle OSI et modèle ATM :

- Couche physique OSI : la couche physique OSI est dans le modèle ATM constituée de la couche physique.
- Couche logique OSI : la couche SAAL du modèle de signalisation respecte toutes les caractéristiques données comme définition de la couche logique : transfert fiable, correction d'erreurs, contrôle de flux et service de synchronisation.
- Couche réseau OSI : la couche de signalisation ATM a comme seul et unique but d'assurer l'établissement, le maintien et la fermeture d'une connexion entre deux ou plusieurs utilisateurs. L'association signalisation/couche réseau peut donc se faire sans risques.

Quant à la couche ATM, il semble bien difficile a priori de la ranger dans l'une ou l'autre des couches du modèle OSI. Indépendante du support physique, elle ne peut être associée à la couche physique OSI. Il semblerait plus juste de l'inclure, avec la couche SAAL, dans la couche logique OSI : la couche ATM traite bien de blocs de données (c'est elle qui fabrique les cellules ATM) et elle implémente (à l'interface UNI uniquement) un système de contrôle de flux (le GFC dans chacune des cellules ATM).

3. Signalisation à l'interface entre utilisateur et réseau

Ce chapitre est dédié à la description du protocole de signalisation UNI 3.1 de l'ATM Forum, utilisé pour les procédures de signalisation entre un TE et son point d'accès au réseau (un commutateur) ainsi que pour celles entre un commutateur et les TE qui y sont connectés.

Un TE désirant se connecter à un autre TE se trouve dans deux cas de figure différents. Ceux-ci seront toutefois totalement transparents pour les deux TE. La différence réside dans le ou les protocoles qui seront utilisés :

1. le TE avec lequel il désire se connecter est branché sur le même point d'accès au réseau que lui. Dans ce cas, le seul protocole utilisé sera le protocole UNI 3.1.
2. le TE avec lequel il désire se connecter est branché sur un autre commutateur, joignable à travers un réseau privé ATM. Dans ce cas, deux protocoles seront utilisés : UNI 3.1 entre le TE appelant et le commutateur sur lequel il est connecté ainsi qu'entre le TE appelé et le commutateur sur lequel il est connecté et PNNI (présenté dans le chapitre suivant) entre tous les commutateurs menant au TE appelé.

L'introduction situera le contexte et les configurations à envisager pour le modèle de signalisation traité dans ce chapitre ainsi que dans le suivant.

Avant toute procédure de signalisation à partir d'une ressource ATM, celle-ci doit d'abord signifier sa présence auprès de son point d'accès au réseau. Cette procédure d'enregistrement d'adresses utilise l'*Interim Local Management Interface* (ILMI) - un protocole défini par l'ATM Forum - et sera exposée dans une première section.

Tout comme SSCOP et SSCF, UNI fonctionne sur base d'une machine à états finis, la réception d'un message provoquant, s'il y a lieu, la transition d'un état à un autre. Après avoir défini l'ensemble des états d'une couche de signalisation UNI, nous définirons les différents types de message utilisés, les éléments d'information (*Information Elements ou IE*) qui les composent et la structure de ces messages.

UNI fonctionne sur le modèle « Request - Indication - Confirmation - Response ». Nous mettrons en évidence les différents types de primitive offerte par UNI au contrôle d'appel - le programme qui va demander l'ouverture de connexion - ainsi que l'utilisation par UNI des primitives offertes par sa couche inférieure (SAAL - SSCF).

Le contrôle d'appel que nous venons d'introduire se situe également dans la couche de signalisation. Celle-ci est subdivisée en deux "modules" : le **contrôle de protocole** et le **contrôle d'appel**, comme l'illustre la Figure 3-1.

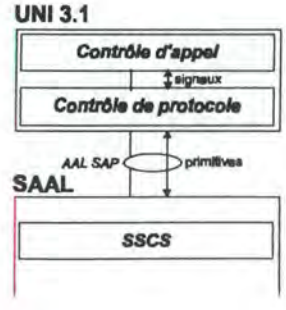


Figure 3-1 : contrôle d'appel et contrôle de protocole

Le contrôle de protocole peut être vu comme le module qui exécute le protocole de signalisation en tant que tel. Le contrôle d'appel est l'utilisateur des services offerts par le contrôle de protocole. Il s'agit d'un programme particulier destiné à gérer les procédures de signalisation sur demande de programmes utilisateur de plus haut niveau. On pourrait ainsi considérer ce contrôle d'appel comme un pilote-gestionnaire de réseau (un *driver*) tels ceux que l'on installe pour le support d'Ethernet sur des stations de travail.

Une dernière section exposera les différents scénarios de signalisation : ouverture, fermeture et redémarrage d'une connexion, tant du point de vue de la ressource ATM (le TE) que de celui de son point d'accès au réseau. Pour certains types d'application, comme l'enseignement à distance ou la vidéoconférence, il peut être nécessaire pour une ressource ATM de se connecter à plusieurs autres ressources ATM (l'ensemble des élèves suivant la même formation dans le cas de l'enseignement à distance ou l'ensemble des correspondants dans le cas de la vidéoconférence). UNI peut être utilisé pour l'ouverture d'une connexion entre deux ressources (point-à-point) ou entre une et plusieurs ressources (point-à-multipoint). Ces deux aspects seront étudiés dans cette même section. Des schémas représentant les flux de messages serviront de supports tout au long de cette section.

3.1 Introduction

Le protocole ATM Forum UNI 3.1 est celui destiné à la signalisation à l'interface entre un équipement terminal ou ressource ATM et son point d'accès à un réseau ATM privé. L'équivalent de ce protocole dans les réseaux ATM publics est le protocole Q.2931, standardisé par l'ITU-T. Il y a peu de différences entre ceux-ci. Le but de ce chapitre n'est pas d'étudier ces différences, mais on se contentera de noter que la principale réside dans l'adressage utilisé : Q.2931 ne supporte que les adresses au format E.164 alors que UNI 3.1 supporte trois modes d'adressage privés : E.164, DCC et ICD (voir la section 1.1.6, page 16).

La Figure 3-2 montre les interfaces où l'on retrouve les protocoles UNI 3.1 et Q.2931. Alors que UNI 3.1 n'est utilisé qu'à l'interface entre un TE et son point d'accès au réseau privé, Q.2931 se trouve à l'interface entre un TE et un point d'accès au réseau public ainsi qu'à l'interface entre un réseau privé et un réseau public. Il incombera donc à un commutateur faisant partie d'un réseau privé d'adapter, si nécessaire, un message de signalisation à destination d'un réseau public au format Q.2931.

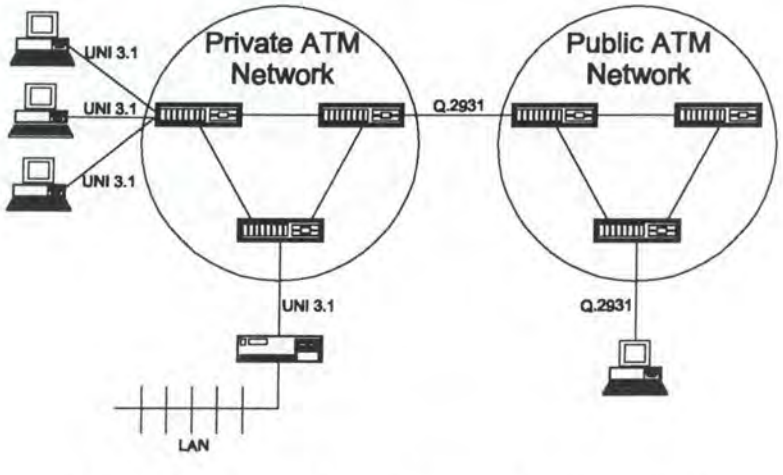


Figure 3-2 : exemple de configuration de réseau ATM

Le protocole UNI 3.1 de l'ATM Forum est constitué de deux parties principales : l'une consacrée à la spécification de l'aspect physique de l'interface UNI, l'autre consacrée à l'aspect signalisation. Ce chapitre exposera exclusivement l'aspect signalisation.

Dans la prochaine section nous présentons brièvement le protocole ILMI résidant à l'interface UNI et utilisé pour l'enregistrement des adresses ATM. Il serait en effet absurde d'enclencher une procédure de signalisation alors qu'un point d'accès au réseau n'est pas encore averti de la présence d'un TE qui y est connecté physiquement.

3.2 Procédure d'enregistrement d'adresses (ILMI)

ILMI est un protocole défini par l'ATM Forum. Il a été conçu afin de compléter les fonctions de management et de maintenance offert par le trafic de cellules OAM (Operation - Administration - Maintenance). On trouvera une brève introduction au trafic OAM en annexe C.

Outre des fonctions de management, de configuration et de maintenance qui ne seront pas exposées ici (ceci nous entraînerait hors du cadre de ce mémoire), ILMI est utilisé afin de permettre à un TE d'enregistrer son adresse auprès de son point d'accès au réseau.

ILMI est constitué de deux composants principaux : SNMP et l'ILMI Management Information Base (ILMI MIB). En toute logique, SNMP est utilisé dans ILMI sans l'adressage IP, non utilisé dans les réseaux ATM.

Une MIB est un concept important dans la gestion de réseaux [BLA95]. Cette base d'informations est constituée d'un ensemble d'objets représentant diverses caractéristiques du réseau qui doivent être gérées, un nom unique étant attribué à chacun des objets afin de l'identifier de manière univoque. On associe à chaque objet de la MIB le mode d'accès (lecture unique, lecture et écriture, ...) permis pour une entité de gestion de chaque TE. La MIB ILMI contient des informations relatives à l'interface physique, à la couche ATM, aux VCC et VPC ainsi qu'aux adresses et préfixes du réseau.

La procédure d'enregistrement d'adresses est effectuée par l'UNI Management Entity (UME). L'UME se situe dans le plan de gestion tant du côté utilisateur de l'UNI que du côté réseau. L'UME utilisateur et l'UME réseau possèdent chacun une MIB ILMI. Dans cette MIB, on trouve, entre autres, deux tables concernées par la procédure d'enregistrement d'adresses :

- la table des adresses : cette table est implémentée par l'UME du côté réseau de l'interface UNI. Son rôle est de maintenir la liste des adresses actives de tous les TE connectés au commutateur concerné.
- la table des préfixes réseau : cette table est implémentée par l'UME du côté utilisateur de l'interface UNI. Son rôle est de maintenir la liste des préfixes réseau (tout champ de l'adresse hormis ESI et SEL) auquel le TE est connecté.

La Figure 3-3 (page 40) expose la procédure appliquée pour l'enregistrement des adresses.

Notons que le message d'initialisation SNMP Cold Start de la Figure 3-3 peut être aussi bien envoyé par le côté utilisateur que par le côté réseau de l'UNI : tout dépend de qui démarre le premier. L'échange de messages se fait par l'intermédiaire d'un canal virtuel entre les deux équipements ATM (le TE et le commutateur).

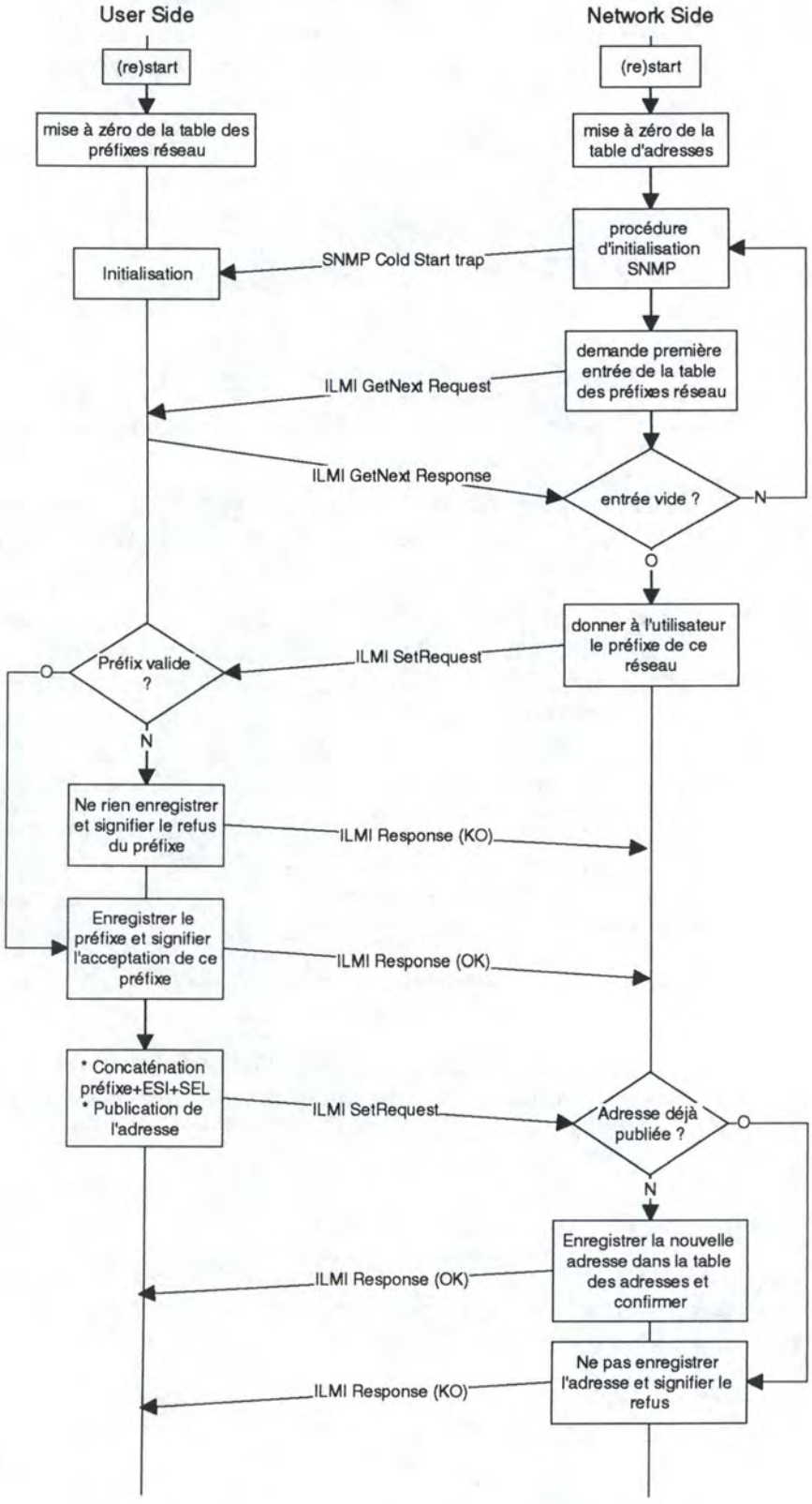


Figure 3-3 : procédure ILMI d'enregistrement d'adresses à l'UNI

Comme nous l'avons dit dans les deux premiers chapitres (section 1.1.4a et section 2.3), certains couples de valeurs VPI/VCI sont réservés pour certains protocoles. ILMI est un de ces protocoles. La

raison pour laquelle un couple VPI/VCI a été réservé pour ILMI est la suivante : supposons qu'un TE vient de se connecter à un point d'accès au réseau; afin de signaler sa présence et enregistrer son adresse, il doit y avoir un échange de messages entre les deux entités ILMI. Deux solutions sont alors envisageables :

1. l'appel à une procédure de signalisation qui aura pour but d'allouer un couple VPI/VCI et de la largeur de bande à l'échange de messages entre entités ILMI. Mais à nouveau, ceci implique l'existence d'un canal virtuel ouvert entre les deux entités de signalisation paires (dans le TE et le commutateur) destiné à l'échange de messages de signalisation;
2. l'allocation au préalable de couples VPI/VCI pour des protocoles de "bas niveau", où le terme "bas niveau" doit être compris ici dans le sens de "premier exécuté" : il y a toujours un protocole de base devant être exécuté avant tous les autres et permettant le fonctionnement des protocoles venant par la suite. Nous verrons dans ce chapitre que le protocole UNI est un protocole de bas niveau : c'est lui qui est exécuté entre deux machines (ou plus) afin d'ouvrir un canal de communication entre ces machines.

ILMI est un protocole de bas niveau : c'est réellement le premier protocole qui est exécuté lorsque l'on connecte un TE à un commutateur. Considérant ceci, on comprend mieux pourquoi un couple VPI/VCI a été commis d'office pour ce protocole. Le couple VPI/VCI associé à ILMI est VPI=0, VCI=16.

L'ensemble des messages SNMP et ILMI ainsi que leurs primitives associées ne seront pas présentés dans ce mémoire.

3.3 ATM Forum UNI 3.1

Un TE étant supposé être connecté à son point d'accès au réseau et la procédure d'enregistrement d'adresses étant supposée terminée et réussie, ce TE peut maintenant démarrer une procédure de signalisation.

Exposons tout d'abord tous les états par lesquels peut passer une entité de signalisation. Ceci nous donnera un point de vue global sur les différents scénarios de signalisation possibles.

3.3.1 Call States

Comme pour la majorité des protocoles de télécommunication, UNI 3.1 peut être représenté par une machine à états finis, chacun des états représentant les conditions de l'un ou l'autre côté de l'interface UNI (TE - point d'accès au réseau) à un moment donné de l'exécution du protocole.

Il s'agit plus spécifiquement de l'état d'une procédure de connexion spécifique vue de chacun des côtés de l'interface (côté utilisateur ou réseau). Il n'est en effet pas possible de définir de manière non ambiguë l'état d'un côté quelconque de l'interface (utilisateur ou réseau), plusieurs procédures d'appel pouvant être exécutées simultanément et pouvant se trouver dans des états différents.

On distingue 3 "classes" d'états en UNI :

1. les états associés au côté utilisateur de l'interface (U);
2. les états associés au côté réseau de l'interface (N);
3. les états associés au concept de "référence globale" : alors que les états de type U et N font référence à une procédure de connexion identifiée vue du côté utilisateur ou réseau de l'interface, la référence globale permet de faire référence à une procédure de connexion spécifique - et ce pour les deux côtés de l'interface - ou à toutes les procédures de connexion en cours sur tous les chemins virtuels contrôlés par l'entité de signalisation.

Comme nous l'avons dit dans l'introduction de ce chapitre, UNI peut également être utilisé pour des procédures de connexion en mode point-à-multipoint. On peut donc ajouter une liste d'états particulière associée aux procédures point-à-multipoint.

3.3.1.a) Etats U et N

Les états U et N reflètent l'évolution d'une procédure de connexion : demande d'ouverture, signification de l'acceptation, demande de fermeture ou signification du refus de connexion, ...

Bien qu'ayant la même dénomination, les états de type U et les états de type N se différencient par leur vision du même événement : chacun voit l'événement de son côté de l'interface.

Le Tableau 3-1 reprend l'ensemble des états U et N ainsi que leur signification. La Figure 3-4 clarifie l'ensemble des termes "côté appelant, côté appelé, utilisateur appelant et utilisateur appelé" repris dans le Tableau 3-1.

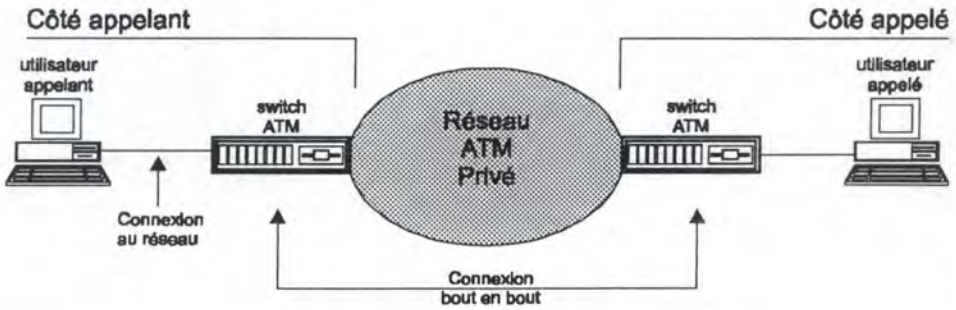


Figure 3-4 : configuration de signalisation UNI

Le Tableau 3-1 nous montre intuitivement différentes phases que l'on va trouver dans un processus de signalisation :

- Les états U1, N1, U3, N3, U6, N6, U9 et N9 suggèrent une phase de demande de connexion d'un TE à un autre et ce par l'intermédiaire des points d'accès au réseau respectifs de ces deux TE.
- Les états U8, N8, U10 et N10 suggèrent une phase d'acceptation de l'appel par le TE appelé.
- Les états U11, N11, U12 et N12 suggèrent une phase de fermeture de connexion.

Nous verrons dans la section 3.3.2 comment ces différentes phases sont répercutées dans les messages, nettement plus explicites que les états.

Dénomination	*Côté U*	*Description*	*Côté N*	*Description*
Null	U0	Pas d'appel en cours	N0	Pas d'appel en cours
Call Initiated	U1	L'utilisateur a fait une demande de connexion au réseau	N1	Le réseau a reçu une demande d'ouverture de connexion mais n'y a pas encore répondu
Outgoing Call Proceeding	U3	L'utilisateur est notifié par le réseau que celui-ci a tous les paramètres nécessaires à l'établissement d'une connexion	N3	Le réseau a spécifié à l'utilisateur qu'il a tous les paramètres nécessaires à l'établissement d'une connexion
Call Delivered	U4	Non supporté dans UNI 3.1	N4	Non supporté dans UNI 3.1
Call Present	U6	L'utilisateur a reçu une demande d'établissement de connexion de la part du réseau mais n'y a pas encore répondu	N6	Le réseau a envoyé une demande d'établissement de connexion à l'utilisateur distant mais n'a pas encore eu de réponse
Call Received	U7	Non supporté dans UNI 3.1	N7	Non supporté dans UNI 3.1
Connect Request	U8	L'utilisateur a accepté une demande connexion et attend confirmation du réseau	N8	Le réseau a reçu une réponse suite à une demande de connexion mais la connexion n'est pas encore allouée
Incoming Call Proceeding	U9	L'utilisateur a envoyé confirmation au réseau de la demande de connexion	N9	L'utilisateur distant a notifié au réseau qu'il a reçu toutes les informations nécessaires pour la demande de connexion
Active	U10	Pour l'utilisateur appelant : confirmation que l'utilisateur distant a accepté la connexion et que celle-ci est disponible Pour l'utilisateur distant : l'utilisateur a reçu confirmation du réseau de la disponibilité de la connexion	N10	Du côté appelé : le réseau a rendu la connexion disponible pour l'utilisateur distant Du côté appelant : le réseau a signalé à l'utilisateur appelant que l'utilisateur distant accepte la connexion
Release Request	U11	L'utilisateur a demandé au réseau de fermer la connexion et attend confirmation	N11	Le réseau a reçu de l'utilisateur une demande de fermeture de connexion
Release Indication	U12	L'utilisateur a reçu une demande de fermeture de connexion de la part du réseau (celui-ci a déjà fermé la connexion bout en bout, pour peu qu'il y en ait une)	N12	Le réseau a fermé la connexion bout en bout et demande à l'utilisateur de fermer sa connexion au réseau

Tableau 3-1 : liste des états utilisateur et réseau à l'interface UNI

3.3.1.b) Référence globale

Le concept de référence globale est uniquement associé à la procédure de redémarrage (se reporter à la section 3.3.4). Cette procédure a pour but de remettre l'état d'une procédure d'appel spécifique ou de toutes les procédures d'appel contrôlées par l'entité de signalisation à l'état nul. Le résultat d'une telle procédure est de rendre le ou les canaux virtuels contrôlés par l'entité de signalisation à nouveau disponibles pour toute procédure de signalisation ultérieure.

Le Tableau 3-2 reprend l'ensemble des états de référence globale associé aux deux côtés de l'interface UNI.

Dénomination	*Label de l'état*	*Description du côté utilisateur*	*Description du côté réseau*
Null	Rest 0	Aucune transaction de référence globale en cours	Aucune transaction de référence globale en cours
Restart Request	Rest 1	L'utilisateur a envoyé une demande de redémarrage au réseau mais celui-ci n'a pas encore envoyé d'acquittement	Le réseau a envoyé une demande de redémarrage à l'utilisateur mais celui-ci n'a pas encore envoyé d'acquittement
Restart	Rest 2	Une demande de redémarrage a été reçue du réseau mais des réponses de toutes les références d'appel n'ont pas encore été reçues	Une requête de redémarrage a été reçue de l'utilisateur mais des réponses de toutes les références d'appel n'ont pas encore été reçues

Tableau 3-2 : liste des états en référence globale

3.3.1.c) Etats pour connexions point-à-multipoint

Le Tableau 3-3 reprend la liste des états s'appliquant lors d'une procédure point-à-multipoint.

Comme nous le voyons dans ce tableau nous pouvons retrouver les phases principales d'une connexion point-à-point : ouverture de connexion avec un nouveau membre d'une connexion point-à-multipoint (états P1 et P2) et retrait d'un membre d'une connexion (état P3 et P4). Les messages pour procédures point-à-multipoint présentés à la section 3.3.2b permettront de mieux imaginer les étapes nécessaires à l'ajout et au retrait d'un TE à une connexion point-à-multipoint. La section 3.3.4c exposera une procédure complète d'ajout d'un membre à une connexion déjà existante.

Dénomination	*Label de l'état*	*Description*
Null	P0	Aucune procédure point-à-multipoint en cours
Add Party Initiated	P1	Un message a été envoyé vers le TE distant afin de l'ajouter à la connexion point-à-multipoint
Add Party Received	P2	Un message a été reçu afin d'ajouter un membre à une connexion point-à-multipoint
Drop Party Initiated	P3	Un message a été envoyé pour supprimer un membre de la connexion point-à-multipoint
Drop Party Received	P4	Un message a été reçu demandant de supprimer un membre de la connexion point-à-multipoint
Active	P5	Un message a été envoyé ou reçu et indique qu'un membre a été ajouté à la connexion point-à-multipoint

Tableau 3-3 : liste des états pour procédures point-à-multipoint

3.3.2 Messages

Les messages sont des informations que deux entités paires de signalisation vont s'envoyer afin de mettre à la disposition d'une application résidant dans le plan utilisateur un canal virtuel pour l'échange de données utilisateur. Ces messages constituent les PDU du protocoles UNI.

A la section 3.3.1a, les états U et N nous avaient montré intuitivement qu'il y avait différentes phases durant la signalisation : demande de connexion, acceptation (ou refus) de la connexion par l'entité de

signalisation appelée et fermeture de connexion. On peut classer les différents messages du protocole UNI 3.1 selon ces trois phases. Nous verrons également par la suite qu'il existe des messages se rapportant à d'autres phases, moins intuitives que celles que nous avions décelées : la phase de demande d'information et la phase de redémarrage. De plus, un TE peut désirer se connecter soit à un seul TE distant - on parlera d'une connexion point-à-point - soit à plusieurs TE distant simultanément - on parlera alors d'une connexion point-à-multipoint.

Dans une première section, nous exposerons les messages utilisés pour la signalisation en mode point-à-point. Une deuxième section présentera les messages spécialement dédiés aux procédures de connexion point-à-multipoint. Une troisième section exposera les messages utilisés dans la procédure de redémarrage. Une quatrième et dernière section nous montrera la structure des messages de signalisation.

A chaque fois que nous décrirons un message nous donnerons les informations transportées par celui-ci dans les scénarios de signalisation les plus généraux (donnés section 3.3.4). Le lecteur désireux d'obtenir une liste exhaustive de ces informations peut se reporter à [UNI3.1-94], [KYAS95] et [BLA95].

Le lecteur ne doit pas perdre de vue qu'avant tout envoi de message par le TE appelant, une connexion offrant un mode de transfert assuré des messages de signalisation doit être mis à la disposition de l'entité de signalisation. Cette connexion se situera entre le TE appelant et son point d'accès au réseau et aura été ouverte par la couche SAAL suite à une demande de l'entité de signalisation.

3.3.2.a) Messages point-à-point

i - Messages impliqués dans la phase de demande de connexion

La première phase que nous avons décelée pour toute procédure de signalisation est la phase de demande de connexion. Deux messages sont utilisés dans cette phase : le message SETUP et le message CALL PROCEEDING.

I. SETUP

Le message SETUP est un message représentant une demande d'ouverture de connexion. Comme nous le verrons par la suite, ce message véhicule un certain nombre d'informations caractérisant le type de connexion souhaité et, bien entendu, l'adresse du TE à qui ce message est destiné.

Le message SETUP est le message envoyé par le TE appelant vers son point d'accès au réseau, ainsi que du point d'accès au réseau de l'ATM TE distant (appelé) au TE appelé. Ce message a alors une signification *globale* : il concerne tous les équipements ATM impliqués dans cette procédure de connexion (TE appelant, point d'accès au réseau côté appelant et appelé et TE appelé) et traverse donc tout le réseau. La Figure 3-5 illustre le concept de signification globale. Notons que le TE appelé pourrait très bien être connecté au même point d'accès au réseau que le TE appelant, la différence résidant seulement dans le fait que le message n'a pas à traverser un réseau ATM. Que l'on se trouve dans une situation ou l'autre n'affecte en rien la logique générale.

Le message SETUP permet l'ouverture d'une connexion unidirectionnelle ou bidirectionnelle. Pour une connexion bidirectionnelle, la largeur de bande que l'utilisateur désire réserver peut être symétrique ou asymétrique, i.e. identique ou différente pour chacune des directions. Par la suite, nous verrons comment l'utilisateur peut spécifier les caractéristiques de sa connexion.

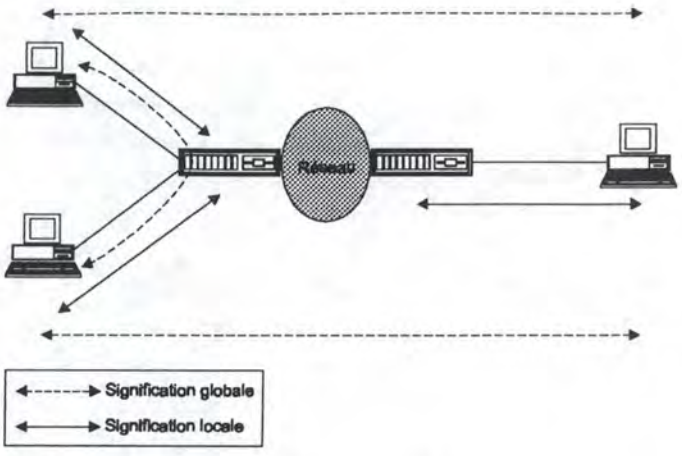


Figure 3-5 : signification globale - signification locale

Un message SETUP transporte les informations suivantes :

- la référence de l'appel : un identifiant unique (le *Call Reference*) est attribué pour chacune des procédures de connexion par le TE appelant et a une signification locale uniquement. Sa durée de vie est équivalente à la durée de la connexion (i.e. jusqu'à ce que celle-ci soit fermée). La référence d'appel est transportée dans tous les messages de signalisation. Nous ne la mettrons plus dans la liste des informations transportées dans les messages ci-après mais le lecteur doit garder en mémoire que cette information est toujours obligatoire.
- l'adresse ATM du TE qui émet la demande de connexion.
- l'adresse ATM du TE avec lequel on désire établir une connexion. On trouvera en annexe D la structure d'encodage de cette information dans le message SETUP.
- le descripteur de trafic ATM : pour rappel, nous avions dit au chapitre 1 que lors d'une ouverture de connexion le TE devait établir un contrat de trafic avec son point d'accès au réseau. Ce contrat de trafic devait reprendre la QoS désirée ainsi qu'un descripteur de trafic source. Le descripteur de trafic ATM contenu dans un message SETUP permet à l'utilisateur de caractériser le trafic qu'il compte générer avec les paramètres décrits à la section 1.1.5b). C'est grâce à cet élément que l'utilisateur peut également définir si la connexion qu'il désire est unidirectionnelle ou bidirectionnelle. On trouvera en annexe D la structure d'encodage de cette information dans le message SETUP.
- la QoS que l'on désire avoir (classe QoS 1, 2, 3, 4 ou non spécifiée - voir section 1.1.5a).
- les paramètres de description AAL : ces paramètres permettent de spécifier le type d'AAL que l'on va employer durant la connexion[2], le type de trafic désiré (émulation de circuit, vidéo, audio haute qualité, ...) ainsi que le taux de transfert en mode CBR (de 64 Kbits/s à 139264 Kbits/s par paliers prédéfinis).
- le *Broadband Bearer Capability* : cet élément d'information contenu dans le message permet de compléter la description de la connexion à l'attention du TE appelé. On y spécifiera entre autres si la connexion désirée est de type point-à-point ou point-à-multipoint, si le trafic est de type VBR ou CBR et si la synchronisation temporelle est nécessaire. On trouvera en annexe D la structure d'encodage de cette information dans le message SETUP.

---

[2] Notons que dans la spécification actuelle de UNI, version 3.1, il n'est possible que de définir les classes A (AAL type 1), B (AAL type 3/4), et C (AAL type 5).

II. CALL PROCEEDING

Le message CALL PROCEEDING est un message d'accusé de réception du message SETUP. Ce message sera donc renvoyé par le point d'accès au réseau du TE appelant à ce TE suite à la réception du message SETUP et envoyé également par le TE appelé à son point d'accès au réseau (suite à la réception d'un message SETUP délivré par ce point d'accès). La Figure 3-8, page 57, illustre l'échange des messages SETUP et CALL PROCEEDING entre deux TE et leur point d'accès au réseau respectif.

Ce message constituant un accusé de réception local, c'est-à-dire relatif uniquement au transfert du message SETUP entre un TE et son point d'accès, est désigné comme ayant une signification locale. Le concept de signification locale est illustré à la Figure 3-5.

Le message CALL PROCEEDING transporte principalement l'identifiant de la connexion (*Connection Identifier*). Cet élément d'information permet d'identifier les ressources ATM utilisées pour la connexion à l'interface entre un TE et son point d'accès au réseau. Il spécifie le couple VCI/VPI à utiliser une fois la connexion établie. Nous verrons à la section 3.3.4 par qui et comment cet identifiant est attribué.

ii - Messages impliqués dans la phase d'acceptation de connexion

La deuxième phase que nous avons décelée pour toute procédure de signalisation est la phase d'acceptation de la connexion. Deux messages sont utilisés dans cette phase : le message CONNECT et le message CONNECT ACKNOWLEDGE.

Suite à l'échange de ces deux messages, un canal virtuel est mis à la disposition de l'utilisateur.

I. CONNECT

Le message CONNECT est le message envoyé par l'utilisateur appelé vers son point d'accès au réseau afin de signaler qu'il accepte la connexion. Ce message sera ensuite transporté à travers le réseau (s'il y a lieu) jusqu'au point d'accès au réseau du TE appelant qui le délivrera à ce TE afin de signaler l'acceptation de la connexion par le TE appelé.

Ce message a une signification globale étant donné le fait qu'il concerne l'ensemble des équipements ATM impliqués dans la procédure de connexion.

Le message CONNECT ne transporte pas d'informations particulières. Toutefois, il peut dans certains scénarios de connexion transporter l'identifiant de connexion.

II. CONNECT ACKNOWLEDGE

Le message CONNECT ACKNOWLEDGE est le message envoyé par le point d'accès au réseau du TE appelé à celui-ci afin de signaler la mise à disposition du canal virtuel pour le trafic utilisateur. Il est également envoyé par le TE appelant à son point d'accès au réseau afin d'acquitter la réception du message CONNECT. La Figure 3-8, page 57, illustre l'échange des messages CONNECT et CONNECT ACKNOWLEDGE.

Ce message a une signification locale uniquement et ne transporte aucun type d'information particulier.

Notons qu'un TE appelé peut très bien décider de refuser une connexion. L'indication de ce refus se fait par l'intermédiaire des mêmes messages que ceux utilisés dans la procédure de fermeture de connexion.

iii - Messages impliqués dans la phase de fermeture de connexion

La troisième phase que nous avons décelée pour toute procédure de signalisation est la phase de fermeture de la connexion. Deux messages sont utilisés dans cette phase : le message RELEASE et le message RELEASE COMPLETE.

Suite à l'échange de ces deux messages, la connexion est fermée et les ressources utilisées (le VC alloué par la phase de demande et d'acceptation de connexion) sont disponibles pour toute autre procédure de signalisation.

I. RELEASE

Le message RELEASE est le message envoyé par un TE désirant fermer une connexion établie avec un autre TE vers son point d'accès au réseau. Ce message sera ensuite transporté à travers le réseau vers le TE participant à la connexion. Ce message est également envoyé par un TE appelé suite à une demande d'ouverture de connexion s'il ne désire pas accepter cette connexion.

Ce message traversant tout le réseau (pour peu que les TE soient connectés à des point d'accès distants) et concernant tous les équipements ATM impliqués dans la connexion a donc une signification globale. Il transporte la cause du refus de la connexion ou la raison pour laquelle une demande de fermeture de connexion a été émise. On trouvera en annexe D l'ensemble des causes pouvant être spécifiées dans un message RELEASE. L'annexe D montrera la structure de cet élément d'information.

II. RELEASE COMPLETE

Le message RELEASE COMPLETE est le message d'acquittement de réception du message RELEASE.

Si le message RELEASE a été envoyé du TE vers son point d'accès au réseau, celui-ci renverra un message RELEASE COMPLETE (après avoir transmis le message RELEASE au réseau ou au TE pour peu que celui-ci soit connecté au même point d'accès).

Si le message RELEASE a été envoyé d'un point d'accès au réseau à un TE qui y est connecté, suite à l'envoi d'un message RELEASE COMPLETE, ce TE n'aura plus aucune connexion avec son point d'accès. Les ressources du réseau seront à nouveau disponibles pour toute demande de connexion.

Le message RELEASE COMPLETE a une signification locale uniquement et ne transporte pas d'informations particulières.

iv - Messages pour les procédures de demande d'information

Outre les phases de demande, d'acceptation et de fermeture de connexion, un TE ou un point d'accès au réseau peut utiliser des messages particuliers - les messages STATUS et STATUS ENQUIRY - afin de demander à l'entité *paire* (le point d'accès au réseau dans le cas d'un TE et vice versa) l'état dans lequel elle se trouve. Cette procédure peut être utilisée afin de vérifier que les deux entités paires sont bien dans des états compatibles : si, par exemple, le TE est en état U1 et que le point d'accès est en état N10, il y a manifestement un problème.

I. STATUS ENQUIRY

Le message STATUS ENQUIRY est le message utilisé afin de demander l'état de l'entité paire de signalisation. Sa signification est uniquement locale et ne transporte pas d'informations particulières.

II. STATUS

Le message STATUS est le message de réponse envoyé suite à la réception d'un message STATUS ENQUIRY. Il transporte donc l'état de l'entité de signalisation paire.

Ce message peut également être utilisé sans avoir reçu au préalable un message STATUS ENQUIRY. Dans ce cas, il est utilisé pour notifier à l'entité paire de signalisation certains cas d'erreurs tels que des éléments d'informations manquant dans des messages, la réception d'un type de message inconnu ou non compatible avec l'état de la procédure de signalisation. On spécifiera donc dans le message STATUS la cause pour laquelle celui-ci a été envoyé. La structure de cet élément d'information est identique à celle de la cause du message RELEASE.

Ce message a une signification locale uniquement.

3.3.2.b) Messages point-à-multipoint

Cette section présente les messages impliqués dans une procédure de connexion point-à-multipoint. Nous verrons cependant dans la section "Ajout d'une feuille" (page 59) que, lors de procédures de signalisation en mode point-à-multipoint, certains messages utilisés dans le mode point-à-point sont également utilisés.

Notons une différence majeure avec une connexion point-à-point : alors que dans ce mode de connexion un message SETUP peut être utilisé pour ouvrir une connexion bidirectionnelle supportant si nécessaire des paramètres de connexion différents pour les deux directions, en mode de connexion point-à-multipoint les connexions sont toutes unidirectionnelles.

Il y a 2 catégories de messages dans le mode point-à-multipoint : les messages destinés à ajouter un TE supplémentaire à une connexion existante et les messages destinés à supprimer un TE d'une connexion point-à-multipoint.

i - Messages pour l'ajout d'un TE à une connexion point-à-multipoint

Deux messages sont définis pour l'ajout d'un TE supplémentaire à une connexion point-à-point ou point-à-multipoint. Soulignons que la version UNI 3.1 de l'ATM Forum n'offre pas le support pour des connexions de type multipoint-à-multipoint. Un TE ajouté à une connexion se situe donc toujours du côté appelé et non appelant d'une procédure de connexion.

I. ADD PARTY

Le message ADD PARTY est défini afin d'ajouter un TE à une connexion existante. Considérant que le TE appelant est la **racine** d'un arbre formé par les liens allant vers tous les ATM TE prenant part à la connexion - chacun des TE appelés étant une **feuille** -, si un TE racine désire avoir une connexion point-à-multipoint cela signifie qu'il devra toujours se connecter à la première feuille par l'envoi d'un message SETUP.

Ajoutons que l'ajout d'une feuille se faisant toujours par l'initiative de la racine, un TE ne peut pas décider seul de se joindre à une connexion point-à-multipoint. Le message ADD PARTY est donc toujours envoyé par la racine.

Comme pour un message de type SETUP, le message ADD PARTY transporte l'adresse de la nouvelle feuille de la connexion point-à-multipoint, l'adresse du TE appelant, les paramètres de description AAL mais, à l'encontre du message SETUP, ne permet pas de spécifier la QoS et un descripteur de trafic source ATM. Ceci peut se comprendre par le fait que le flot d'informations va toujours de la racine vers l'ensemble des feuilles; les informations transmises vers chacune des feuilles sont identiques et sont donc toutes caractérisées par le même type de QoS et de descripteur de trafic ATM.

Ce message transporte également une référence de point terminal (*Endpoint Reference*) permettant d'identifier de manière univoque une feuille de la connexion point-à-multipoint. La première feuille de la connexion a toujours la valeur 0.

Ce message a une signification globale.

II. ADD PARTY ACKNOWLEDGE

Le message ADD PART ACKNOWLEDGE est utilisé afin de signaler au TE appelant que l'ajout de la nouvelle feuille a réussi. On peut le comparer à un message CONNECT dans les connexions point-à-point. Il est envoyé par la feuille suite à la réception d'un message ADD PARTY.

Ce message a une signification globale et transporte la même référence de point terminal que celui spécifié dans le message ADD PARTY.

III. ADD PARTY REJECT

Le message ADD PARTY REJECT est utilisé afin de signaler au TE appelant que l'ajout de la nouvelle feuille a échoué. Il transporte entre autres la cause de cet échec. Il est envoyé par la feuille suite à la réception d'un message ADD PARTY.

Ce message a une signification globale et transporte la même référence de point terminal que celui spécifié dans le message ADD PARTY. Il transporte de plus une cause indiquant la raison du refus.

ii - Messages pour le retrait d'un TE à une connexion point-à-multipoint

Deux messages sont définis pour le retrait d'une feuille à une connexion point-à-multipoint : DROP PARTY et DROP PARTY ACKNOWLEDGE.

I. DROP PARTY

Le message DROP PARTY est envoyé par une feuille de la connexion. Il a pour but de retirer cette feuille de la connexion point-à-multipoint en cours, sans toutefois altérer l'état des connexions restantes entre la racine et les autres feuilles. Si toutefois une feuille ou la racine désirait supprimer l'ensemble de la connexion, un message RELEASE serait envoyé. La cause de la fermeture de connexion est toujours indiquée dans un message DROP PARTY.

Ce message a une signification globale (entre la racine et cette feuille). Il transporte la cause associée au retrait de la feuille ainsi que la référence de point terminal associé à la feuille concernée par ce message.

II. DROP PARTY ACKNOWLEDGE

Le message DROP PARTY ACKNOWLEDGE est envoyé comme confirmation suite à la réception d'un message DROP PARTY.

Ce message a une signification locale et transporte la référence de point terminal.

3.3.2.c) Messages pour les procédures de redémarrage

Nous avons vu dans la section 3.3.1b " Call States : Référence globale" qu'il existait 3 états associés à une procédure particulière : la procédure de redémarrage. Nous y avions donné une brève description de cette procédure. Celle-ci sera expliquée plus en détail à la page 61. Les messages associés à cette procédure sont les messages RESTART, RESTART ACKNOWLEDGE et STATUS (ce message étant identique au message STATUS des procédures point-à-point, il ne sera pas réexposé ici).

I. RESTART

Le message RESTART est utilisé par un TE ou un point d'accès au réseau afin de demander à l'entité paire (le point d'accès au réseau dans le cas d'un TE et vice versa) de débuter une procédure de redémarrage.

Ce message a une signification locale.

Le message RESTART doit transporter un indicateur de redémarrage (*Restart Indicator*) permettant de spécifier si la procédure de redémarrage doit s'appliquer à une procédure de signalisation spécifique ou à toutes celles en cours. Dans le cas où la procédure de redémarrage s'applique à une procédure de signalisation spécifique, celle-ci est identifiée dans le message par l'intermédiaire d'un élément d'information de type identifiant de connexion.

II. RESTART ACKNOWLEDGE

Le message RESTART ACKNOWLEDGE est utilisé pour accuser réception du message RESTART.

Il a une signification locale. Il transporte l'indicateur de redémarrage et l'identifiant de connexion si celui-ci a été spécifié dans le message RESTART reçu.

3.3.2.d) Structure des messages UNI

Après avoir présenté tous les messages susceptibles de prendre part dans une procédure de signalisation, nous descendons encore d'un degré de détail et présentons la structure de ces messages.

Tous les messages ou PDU UNI - qu'ils se réfèrent à une procédure point-à-point, point-à-multipoint ou encore à celle de redémarrage - partagent la même structure. Ils sont constitués de deux "blocs" principaux : un en-tête de message permettant d'identifier le type du message ainsi que la procédure de signalisation à laquelle il se rapporte et une partie informationnelle transportant tous les éléments d'information caractérisant le message. Nous avons donné à la section précédente les éléments d'information principaux transportés dans chacun des messages. Le bloc informationnel peut cependant être nul : nous avons vu en effet que certains messages tels que le CONNECT ACKNOWLEDGE ou le RELEASE COMPLETE ne transportaient aucun type d'information particulière.

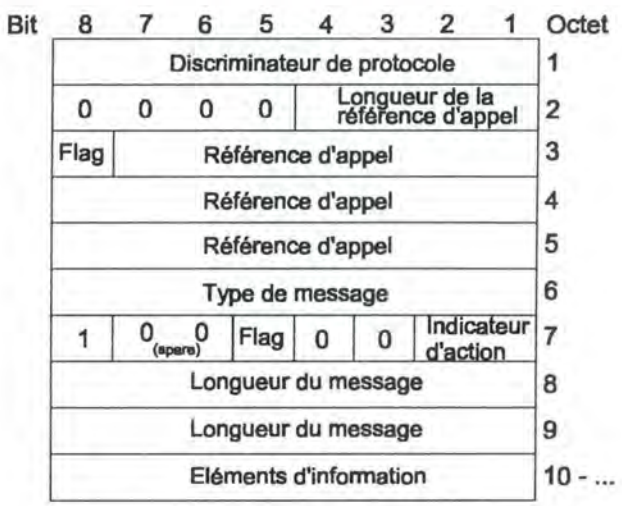


Figure 3-6 : structure d'un message UNI

La Figure 3-6 présente la structure d'un message UNI. L'en-tête est constitué des 9 premiers octets du message et le bloc informationnel comprend tous les octets suivants.

i - En-tête d'un message UNI

L'en-tête de message est constitué d'un certain nombre de champs décrits ci-dessous permettant d'identifier un message UNI (de quel type de message s'agit-il ?), la version du protocole auquel il se rapporte ainsi que la procédure de signalisation à laquelle il fait référence (comme nous l'avons déjà dit, plusieurs procédures de signalisation peuvent être gérées simultanément par l'entité de signalisation).

Reprenons ces éléments un à un :

- le discriminateur de protocole (PD : *Protocol Discriminator*) : le PD permet à l'entité réceptrice de savoir à quelle version du protocole de signalisation elle a affaire. En effet, il n'est pas obligatoire de se restreindre à un seul protocole de signalisation : une entité de signalisation peut, comme nous le verrons par exemple dans le chapitre suivant) être capable de gérer différents protocoles de signalisation à la fois tels UNI 3.1, PNNI, IISP (un protocole de signalisation précurseur à PNNI), Q.2931 (l'équivalent UNI 3.1 de l'ITU-T), etc. Pour UNI 3.1, ce champ a une valeur binaire égale à 00001001.

- la référence d'appel (*Call Reference*) : nous avions déjà introduit le concept de référence d'appel lors de la description du message SETUP. Nous avions dit que cet identifiant était unique et permettait d'identifier de manière univoque une procédure d'appel. Cet identifiant a une signification locale uniquement : cette référence peut donc être différente du côté appelant et du côté appelé du réseau (en rapport avec la Figure 3-4 page 42). La valeur de cet identifiant est codée sur trois octets (le bit le plus significatif est le 7ème bit du premier octet et le bit le moins significatif est le 1er bit du troisième octet) et est attribuée par le TE appelant et par le point d'accès au réseau du côté appelé du réseau. Il n'existe que pendant toute la durée de la phase de signalisation (donc jusqu'au dernier message RELEASE ou RELEASE COMPLETE destiné à fermer une connexion).

  Un indicateur (le *flag* du 1er bit de la référence d'appel) permet de déterminer qui a alloué la référence d'appel : si cette valeur est égale à 0, alors c'est l'entité qui a envoyé le message qui a alloué la référence. Si cette valeur est 1, alors ce n'est pas cette entité qui a alloué la référence. Concrètement lorsque le TE appelant enverra un message SETUP à son point d'accès au réseau, ce flag sera à 0 (ce qui est conséquent avec ce que nous avons dit plus haut : c'est le TE qui alloue la référence du côté appelé du réseau). De même, lorsque le point d'accès transmettra le message SETUP au TE appelé, il mettra le flag à la valeur 0. Par contre, lorsque le TE appelé enverra un message à son point d'accès ou que le point d'accès côté appelant transmettra un message au TE appelant ce flag sera mis à 1, car ni le TE appelé, ni le point d'accès côté appelant n'ont alloué cette référence. Grâce à ce flag, la même référence d'appel peut être allouée des deux côtés de l'interface UNI (par le TE et par son point d'accès au réseau) pour des procédures de connexion différentes mais concurrentes (ayant lieu en même temps).

  Enfin, ajoutons que la référence d'appel peut prendre une valeur particulière, la valeur 0. Dans ce cas, il s'agira de la référence globale (nous avons déjà vu le concept de référence globale à la section 3.3.1b).

- le type de message : le type de message est utilisé afin d'identifier un message envoyé. On trouvera les valeurs binaires identifiant ces messages dans le Tableau 3-4. Remarquons l'octet suivant le type de message et plus particulièrement les 2 bits de l'indicateur d'action : il permet d'indiquer à l'entité réceptrice l'action qui doit être prise si le message n'est pas reconnu (l'entité émettrice pourrait par exemple faire tourner des versions moins récentes du protocole et ne pas reconnaître un nouveau type de message comme valide). L'indicateur d'action dit à l'entité réceptrice si elle doit simplement effacer le message qu'elle a reçu et ne pas en tenir compte ou si elle doit l'effacer mais générer une erreur (i.e. envoyer un message STATUS). Le flag du 7ème octet permet de dire à l'entité réceptrice si elle doit ignorer le champ indicateur d'action ou en tenir compte obligatoirement.

Message	*Identifiant*
SETUP	00000101
CALL PROCEEDING	00000010
CONNECT	00000111
CONNECT ACKNOWLEDGE	00001111
RELEASE	01001101
RELEASE COMPLETE	01011010
RESTART	01000110
RESTART ACKNOWLEDGE	01001110
STATUS	01111101
STATUS ENQUIRY	01110101
ADD PARTY	10000000
ADD PARTY ACKNOWLEDGE	10000001
ADD PARTY REJECT	10000010
DROP PARTY	10000011
DROP PARTY ACKNOWLEDGE	10000100

Tableau 3-4 : liste des identifiants des messages UNI

- la longueur du message : ce champ permet d'indiquer la taille (en octets) du message. Toutefois il ne s'agit pas de la taille globale du message mais de la taille du bloc informationnel. On n'y compte donc pas le PD, la référence d'appel, le type du message ni l'octet contenant le champ d'indicateur d'action ainsi que les 2 octets contenant la longueur du message.

ii - Le bloc informationnel

Le bloc informationnel contient toutes les informations qui sont nécessaires afin de caractériser les messages. Comme nous l'avons déjà dit ce bloc peut très bien être nul. Dans ce cas, le message ne sera composé que de l'en-tête.

Dans le cas où un message doit véhiculer des informations, celle-ci sont placées dans des structures appelées éléments d'information (*Information Element : IE*), illustrés à la Figure 3-7. Chaque IE ne peut être représenté qu'une fois au plus dans un même message. Si plus d'une occurrence d'un IE apparaît, celle-ci est ignorée.

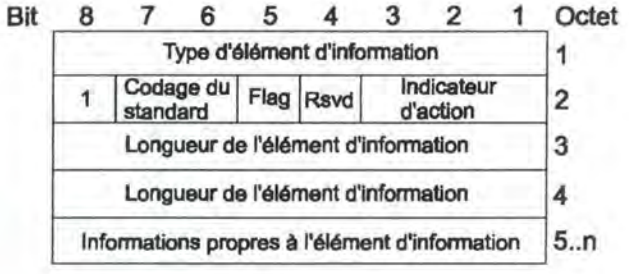


Figure 3-7 : structure d'un IE

Tous les IE ont une structure de base commune constituée des 4 premiers octets de la Figure 3-7. Ils possèdent également des structures propres à chacun des IE, codés à partir du 5ème octet. On trouvera en annexe D des exemples d'IE (descripteur de trafic ATM, adresse du TE appelée, Broadband Bearer Capability, ...) décrivant précisément le codage de toutes les informations qu'ils contiennent.

A l'encontre des messages UNI où le bloc informationnel pouvait être nul, il n'existe pas d'IE ayant une structure propre nulle, c'est-à-dire constituée uniquement de la base commune.

Reprenons les différents éléments constituant la base commune d'un IE :

- le type d'élément d'information : ce champ permet d'identifier univoquement un IE. On trouvera en annexe D un tableau reprenant l'ensemble de ces identifiants.
- le codage du standard : ce champ permet de définir le codage qui a été utilisé afin d'encoder les IE. Dans la majorité des cas, on spécifiera un codage 'ITU-T' (codé 00). Cependant on peut très bien décider d'utiliser son propre codage (codé 11). Dans ce cas, la définition de celui-ci doit pouvoir être trouvée sur le point d'accès au réseau par tous les TE qui y sont connectés.
- l'indicateur d'action : cet indicateur joue le même rôle que l'indicateur d'action dans la structure d'un message UNI donnée à la Figure 3-6 : il permet de dire à l'entité réceptrice d'un message contenant cet IE ce qu'elle doit faire si elle ne le reconnaît pas (le jeter et l'ignorer, le jeter et signaler une erreur via un message STATUS ou encore ignorer l'ensemble du message et signaler une erreur). Le flag du 2ème octet a le même rôle que dans la structure d'un message UNI, c'est-à-dire signaler si l'indicateur d'action doit être ignoré ou s'il faut obligatoirement en tenir compte.
- la longueur de l'élément d'information : ce champ reprend la taille de la structure propre de l'IE (il ne compte donc pas les 4 premiers octets).

3.3.3 Primitives de service

La couche de signalisation (i.e. le contrôle de protocole et le contrôle d'appel) n'agit pas de sa propre initiative. Il doit exister une entité de niveau supérieur qui va demander à la couche de signalisation de lui mettre à sa disposition un canal virtuel. Quelle est cette entité ? Tout simplement toute application de

l'utilisateur qui a besoin d'obtenir un canal virtuel pour transférer des informations. Il pourrait s'agir d'une application de transfert de données, d'une application de téléphonie, de vidéoconférence, ... Les choix sont vastes, surtout avec ATM !

Quelle que soit cette entité de haut niveau, elle doit pouvoir effectuer les demandes d'ouverture et de fermeture de connexion auprès de la couche de signalisation. Tel que nous l'avons introduit, il doit donc exister un pilote ou gestionnaire de réseau ATM constituant le **contrôle d'appel** qui a accès aux primitives offertes par le **contrôle de protocole**. Lorsque ce pilote détecte qu'une application de l'utilisateur demande l'ouverture d'une connexion avec une application s'exécutant dans un TE distant, il demandera au contrôle de protocole d'ouvrir cette connexion à l'aide des primitives que celle-ci lui offre.

La présentation de ces primitives dans cette section suit le développement logique que nous voulons présenter dans ce chapitre : nous avons tout d'abord donné une vue globale des différentes procédures de signalisation par l'intermédiaire des états associés à une procédure de connexion; nous avons ensuite précisé ces procédures en présentant les messages qui entraînaient un changement des états de la procédure de signalisation; les primitives nous montreront enfin comment ces messages sont générés.

La spécification du protocole UNI 3.1 de l'ATM Forum ne définit pas de primitives de services. Les noms des primitives données dans les sections suivantes ont été choisis arbitrairement par l'auteur de ce mémoire et sont déduits du déroulement logique d'une procédure de signalisation.

Toutes les primitives présentées dans cette section génèrent chacune un type de message particulier. Ce message est ensuite passé par l'entité de signalisation à la couche SAAL par l'intermédiaire de la primitive SSCF AAL-MESSAGE-FOR-TRANSMISSION.request(MU) où MU est le message généré par la primitive de signalisation. De même, la couche SAAL délivre un message à la couche de signalisation par le biais d'une primitive SSCF AAL-RECEIVED-MESSAGE.indication(MU).

Le Tableau 3-5 reprend l'ensemble des primitives offertes par le contrôle de protocole au contrôle d'appel. La section suivante exposera des scénarios de signalisation et mettra en lumière l'utilisation de ces primitives.

Notons toutefois que l'utilisation d'une primitive telle que SIG-OPEN peut ne pas refléter les 4 possibilités request, indication, response et confirmation. En effet, nous avons vu qu'un utilisateur peut ne pas vouloir accepter une demande de connexion et signaler son refus par l'envoi d'un message RELEASE. Ce message ne sera pas généré par un SIG-OPEN.resp mais par un SIG-CLOSE.req.

Nom	*Request*	*Indication*	*Response*	*Confirmation*	*Messages generes*
SIG-OPEN	adresse TE appelé, QoS, paramètres AAL, descripteur de trafic ATM, Broadband Bearer Capability	adresse TE appelant[3], QoS, paramètres AAL, descripteur de trafic ATM, Broadband Bearer Capability	Identifiant de connexion	Identifiant de connexion	SETUP, CALL PROCEEDING
SIG-ACCEPT	pas de paramètres particuliers	pas de paramètres particuliers	pas de paramètres particuliers	pas de paramètres particuliers	ACCEPT, ACCEPT ACK
SIG-CLOSE	cause	cause	pas de paramètres particuliers	pas de paramètres particuliers	RELEASE, RELEASE COMPLETE
SIG-ADD_PARTY	adresse TE appelé, référence de point terminal, paramètres AAL	adresse TE appelant[3], référence de point terminal, paramètres AAL	référence de point terminal, cause[4]	référence de point terminal, cause[4]	ADD PARTY, ADD PARTY ACK / ADD PARTY REJECT
SIG-DROP_PARTY	référence de point terminal, cause	référence de point terminal, cause	référence de point terminal	référence de point terminal	DROP PARTY, DROP PARTY ACK
SIG-RESTART	indicateur de redémarrage, identifiant de connexion	indicateur de redémarrage, identifiant de connexion	indicateur de redémarrage, identifiant de connexion	indicateur de redémarrage, identifiant de connexion	RESTART, RESTART ACK

Tableau 3-5 : primitives de service offertes par la couche de signalisation

3.3.4 Scénarios

Nous exposons dans cette section quatre scénarios de signalisation : une ouverture de connexion (demande et acceptation) en mode point-à-point, une fermeture de connexion, l'ajout d'une feuille à une connexion point-à-multipoint et une procédure de redémarrage.

Avant d'utiliser le protocole UNI 3.1 pour ces différents scénarios, il faut toutefois songer à un ensemble de procédures qui doivent être traitées avant tout. Nous les présentons dans leur ordre d'exécution :

1. Initialisation du protocole UNI : cette phase correspond au démarrage de l'entité de contrôle de protocole.
2. Configuration du protocole : suite au démarrage de l'entité de contrôle de protocole, des procédures de configuration devraient être exécutées, telles que :
    - réservation de la mémoire nécessaire à l'exécution des procédures de signalisation. Cette réservation de mémoire peut être nécessaire afin de réserver des tampons de taille suffisante pour la réception et l'envoi de messages et pour le fonctionnement interne de l'entité.
    - configuration du SAP entre la couche SAAL et la couche de signalisation (plus particulièrement, le contrôle de protocole). Nous avons dit en effet au début du chapitre 2 que les couches du modèle de signalisation communiquaient entre elles par l'intermédiaire de

[3] Paramètre non obligatoire

[4] La cause est incluse si l'ajout de la feuille est refusé

primitives à travers un SAP. Il y a un SAP par lien logique (i.e. canal virtuel de signalisation) que UNI contrôle. Notons que pour le protocole UNI, il y a un seul lien logique, celui identifié par VPI=0 et VCI=5. Nous verrons dans le chapitre 4 que pour le protocole PNNI, tous les VC d'identifiant VCI=5 dans tous les VP peuvent également être utilisés pour le transport de messages de signalisation. Configurer ce SAP signifie préciser des valeurs telles que les VCI et VPI à utiliser pour le canal virtuel de signalisation, le type d'interface que UNI représente (TE ou point d'accès au réseau), l'intervalle de valeurs VPI/VCI à affecter à une procédure de connexion (le VPI/VCI que l'utilisateur ou contrôle d'appel pourra utiliser pour le transfert de données).

- configuration du SAP entre le contrôle de protocole et le contrôle d'appel. Il y un SAP par application (et donc contrôle d'appel) utilisant les services de UNI.

3. Connexion aux SAP : une fois que les SAP ont été configurés, on passe par une phase de connexion. Cette phase signifie simplement que les liens (SAP) entre les couches SAAL, contrôle de protocole et contrôle(s) d'appel sont activés.
4. Activation de la connexion : l'activation du SAP entre l'entité UNI et SAAL (SSCF) entraîne automatiquement l'ouverture du lien logique entre les entités SAAL paires. L'activation s'effectue par la primitive AAL-START.request utilisée par l'entité UNI auprès de SSCF. Une fois la connexion établie, SSCF génère un AAL-IN_SERVICE.indication. Le lien logique est alors ouvert et prêt à l'envoi de messages de signalisation.

Suite à l'exécution de ces procédures, l'entité UNI est considérée comme prête pour l'exécution de toute procédure de signalisation.

Le premier service demandé par le contrôle d'appel sera en toute vraisemblance une demande d'ouverture de connexion.

3.3.4.a) Ouverture de connexion

La Figure 3-8 expose le flux de messages entre deux TE et leur point d'accès au réseau pour un phase de demande d'ouverture de signalisation ainsi que les changements d'état associés à la procédure de connexion. Cette figure servira de support à l'exposé qui suit. Les boîtes intitulées T303, T310, etc. correspondent à des timers qui sont activés suite à l'envoi de messages particuliers. Ils sont donnés ici dans un pur but de complétude et sont exposés en annexe E.

Dans cette figure, un TE A désire ouvrir une connexion avec un TE B. Pour demander l'ouverture de cette connexion, le contrôle d'appel du TE A génère un SIG-OPEN.req auprès de l'entité UNI (le contrôle de protocole) en fournissant tous les paramètres nécessaires (se reporter au Tableau 3-5 pour une liste des paramètres à fournir). Suite à cette requête, un message SETUP est envoyé vers l'entité UNI du point d'accès au réseau (le nœud A). Suite à un problème de transmission, le message n'a pu être reçu. Ne recevant pas de réponse suite à l'envoi du SETUP, l'entité UNI renvoie le message. La réception de ce message génère un SIG-OPEN.ind auprès du contrôle d'appel du nœud.

Le contrôle d'appel d'un nœud est un cas assez particulier. Il ne s'agit évidemment pas d'un programme de type utilisateur se déroulant par exemple de manière interactive. Il est en effet inconcevable d'avoir un opérateur humain à côté du nœud dont le seul rôle serait d'acquitter les messages entrant dans ce nœud. Nous verrons plus en détail dans le chapitre 4 ce que pourrait être le contrôle d'appel dans le cas d'un nœud, mais supposons en attendant qu'il s'agit d'un programme qui a pour but d'automatiser le déroulement du protocole UNI. Son rôle serait de gérer les ressources disponibles et d'assurer le transfert à travers le réseau, par l'intermédiaire d'un autre protocole (PNNI), du message qui lui aurait été confié.

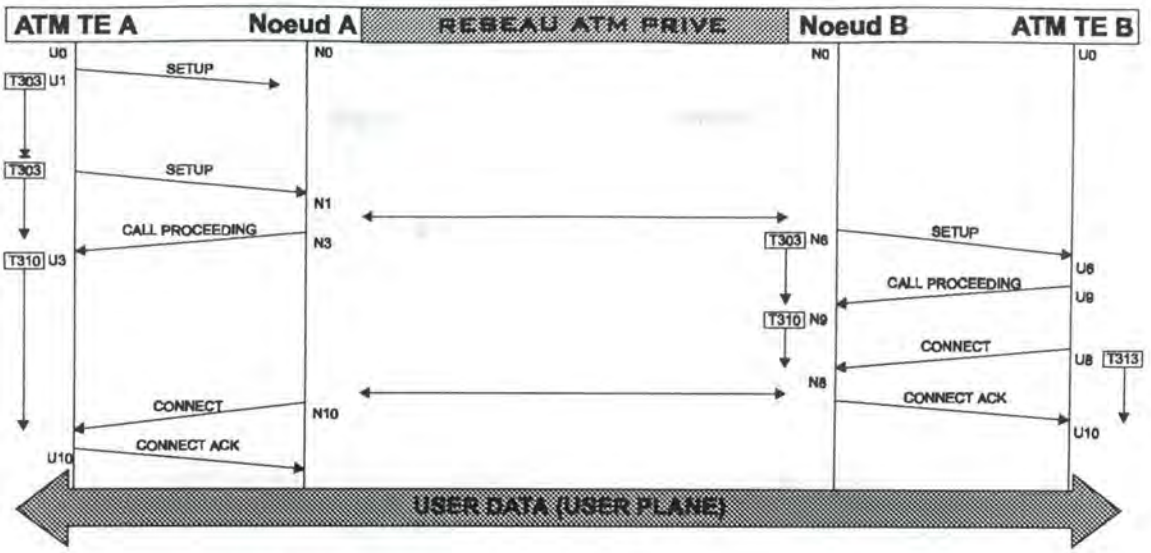


Figure 3-8 : flux de message pour une phase d'ouverture de connexion

Suite à l'indication de réception du message SETUP, le contrôle d'appel du nœud A vérifie s'il peut assurer le service demandé et décrit par le descripteur de trafic ATM et par la classe de QoS. Nous supposons que les ressources nécessaires sont libres. Le contrôle d'appel doit alors assigner un couple de valeurs VPI/VCI libre que le contrôle d'appel utilisateur (dans le TE A) utilisera pour le transfert d'informations utilisateur. Si le contrôle d'appel du nœud A peut envoyer le message de demande de connexion (SETUP) à travers le réseau vers le point d'accès au réseau du TE B (le nœud B), il génère une primitive SIG-OPEN.resp auprès de l'entité UNI en fournissant comme paramètre l'identifiant de connexion constitué du couple VPI/VCI alloué. Ceci provoque l'envoi d'un message CALL PROCEEDING vers le TE A. La réception de ce message au TE A génère un SIG-OPEN.conf auprès du contrôle d'appel en fournissant l'identifiant de connexion alloué par le réseau.

Pendant ce temps, le message SETUP a traversé le réseau et est arrivé au nœud B. Le contrôle d'appel de ce nœud vérifie, tout comme pour le nœud A, s'il peut supporter le service demandé. Dans l'affirmative, il alloue un couple de valeurs VPI/VCI et génère un SIG-OPEN.req auprès de l'entité UNI en spécifiant les paramètres reçus dans le message SETUP ayant traversé le réseau et en y ajoutant l'identifiant de connexion alloué. Cette primitive entraîne l'envoi d'un message SETUP vers l'entité B.

La réception du message SETUP au TE B est signalée au contrôle d'appel par l'entité UNI par une primitive SIG-OPEN.ind. Si le couple VPI/VCI spécifié dans l'identifiant de connexion n'a pas encore été alloué du côté du TE, une primitive SIG-OPEN.resp est utilisée afin d'acquitter la réception du message SETUP et de provoquer l'envoi d'un message CALL PROCEEDING vers le nœud B. Celui-ci n'entreprendra aucune action particulière si ce n'est l'attente d'un message d'acceptation de la connexion.

L'acceptation de la connexion par l'utilisateur B est signalée à l'entité UNI par une primitive SIG-ACCEPT.req provoquant l'envoi du message CONNECT vers le nœud B. La réception de ce message est signalée par la primitive SIG-ACCEPT.ind. A ce moment, le nœud B envoie le message CONNECT à travers le réseau vers le nœud A et acquitte la réception du message CONNECT par la primitive SIG-ACCEPT.resp. Ceci génère l'envoi d'un message CONNECT ACKNOWLEDGE vers le TE B. La réception de ce message, signalée par la primitive SIG-ACCEPT.conf, indique au contrôle d'appel que la connexion est maintenant mise à la disposition de l'utilisateur.

Pendant ce temps, le message CONNECT a traversé le réseau et est arrivé au nœud A. Celui-ci signale la mise à disposition de la connexion au contrôle d'appel du TE A en générant une primitive SIG-ACCEPT.req. Un message CONNECT est alors envoyé au TE A, sa réception générant un SIG-ACCEPT.ind. Le contrôle d'appel acquitte la réception de ce message par un SIG-ACCEPT.resp,

entraînant l'envoi d'un message CONNECT ACKNOWLEDGE vers le nœud A. Suite à l'envoi de ce message, l'utilisateur a accès à la connexion qui vient d'être ouverte. La réception du message CONNECT au nœud B est signalée par la primitive SIG-ACCEPT.conf. Le nœud n'effectue aucune action particulière suite à cette primitive.

En fin de procédure, un canal virtuel est à la disposition du trafic utilisateur entre les TE A et B.

3.3.4.b) Fermeture de connexion

La Figure 3-9 expose le flux de messages entre deux TE et leur point d'accès au réseau pour une phase de fermeture de connexion. Cette figure servira de support à l'exposé qui suit.

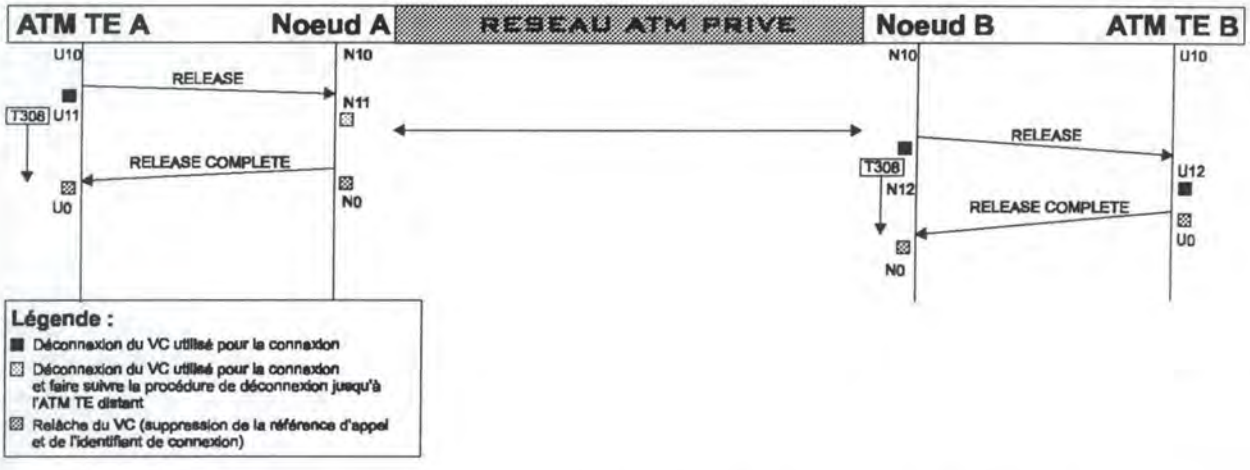


Figure 3-9 : flux de messages pour une procédure de fermeture de connexion

Dans cette procédure, le TE A souhaite fermer la connexion qu'il a avec le TE B. Pour ce faire, le contrôle d'appel génère un SIG-CLOSE.req auprès de l'entité UNI. La cause de fermeture de connexion est insérée par l'entité UNI et précise une fin de connexion normale (voir tableau des valeurs de cause en annexe D). Cette primitive génère l'envoi d'un message RELEASE vers le nœud A. Suite à l'envoi de ce message, l'entité UNI peut déconnecter le VC utilisé pour la connexion. Un VC se trouvant dans l'état déconnecté est un VC ne participant plus à un VCC mais qui n'est pas encore disponible pour une nouvelle procédure de connexion.

La réception de ce message génère une primitive SIG-CLOSE.ind. Suite à l'indication de demande de fermeture, le contrôle d'appel prend les mesures nécessaires afin de déconnecter le VC utilisé pour le transfert d'informations utilisateur entre le TE A et le nœud ainsi que pour fermer la connexion traversant le réseau (la demande fermeture de connexion est alors transportée à travers tout le réseau et arrive jusqu'au nœud B). Le contrôle d'appel demande ensuite à l'entité UNI d'envoyer un message d'acquittement de fermeture de connexion RELEASE COMPLETE à l'aide de la primitive SIG-CLOSE.resp. Une fois que ce message envoyé l'entité UNI peut rendre la référence d'appel à nouveau disponible pour toute procédure de demande de connexion. et le couple VPI/VCI réservé pour la connexion est libéré (le VC peut donc être utilisé pour une autre procédure de connexion).

Une primitive SIG-CLOSE.conf signale au contrôle d'appel du TE A la réception du message RELEASE COMPLETE acquittant le demande de fermeture de la connexion. Suite à la réception de ce message, l'entité UNI peut rendre la référence d'appel et le couple VCI/VPI à nouveau disponible pour toute procédure de connexion.

Pendant ce temps, la demande de fermeture de connexion est arrivée au nœud B. Un échange de messages similaire à celui qui a eu lieu entre le TE A et le nœud A se passe entre le nœud B et le TE B, comme l'illustre la Figure 3-9.

3.3.4.c) Ajout d'une feuille

La Figure 3-10 expose le flux de messages entre deux TE et leur point d'accès au réseau respectif pour la création d'une connexion point-à-multipoint ainsi que le flux de messages engendré par l'ajout d'une feuille à une connexion point-à-multipoint.

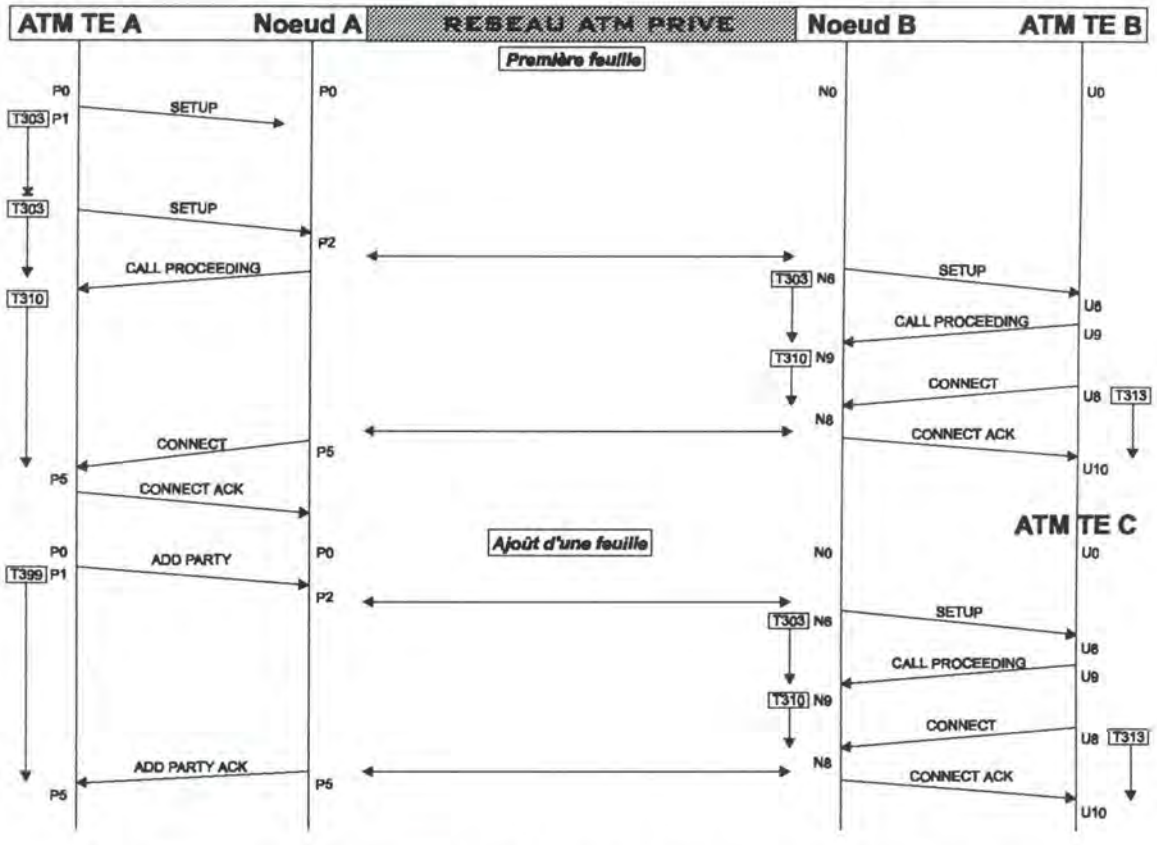


Figure 3-10 : flux de messages pour la création d'une connexion point-à-multipoint

La procédure de création d'une connexion point-à-multipoint, en ce qui concerne la connexion avec la première feuille, est similaire à la création d'une connexion point-à-point entre deux TE. Cependant, le message SETUP envoyé suite à un SIG-OPEN.req doit obligatoirement contenir un IE de type référence de point terminal. Pour la première feuille, la valeur de cette référence doit être égale à 0. D'autre part, l'IE *Broadband Bearer Capability* doit contenir l'indication qu'il s'agit d'une connexion point-à-multipoint (se référer à l'annexe D pour le codage de cet élément). Ajoutons enfin que l'on ne peut spécifier des paramètres de trafic ATM et une classe QoS que pour le sens racine→feuille. Si cependant le message SETUP contenait des informations caractérisant le trafic dans le sens feuille→racine, l'appel serait rejeté.

Les états donnés à la Figure 3-10 (P0, P1, etc.) sont ceux associé à une procédure point-à-multipoint, tels que nous les avons définis à la page 44. Les états correspondant aux procédures d'appels "classiques" (U0, U1, N6, etc.), bien que non représentés sur cette figure, s'appliquent également et simultanément aux états des procédures point-à-multipoint durant l'échange des messages SETUP, CALL PROCEEDING, etc. Toutefois ces états n'existent pas lors d'échange de messages propres aux

procédures point-à-multipoint (ADD PARTY, ADD PARTY ACK). Dans le cas de connexion point-à-multipoint, on parlera alors d'état du *lien* pour les états U0, U1, etc. et d'état de la *procédure point-à-multipoint* pour les états P0, P1, ..., P5.

Supposons qu'une connexion existe déjà entre la racine (le TE A) et la première feuille (le TE B) et que le TE A désire ajouter à la connexion un TE C. Le contrôle d'appel de la racine génère un SIG-ADD_PARTY.req auprès de l'entité UNI en spécifiant l'adresse de la feuille à ajouter à la connexion ainsi que sa référence de point terminal. Le descripteur de trafic ATM, la classe QoS et le *Broadband Bearer Capability* ne sont pas donnés en argument, leur valeur étant obligatoirement identique à celle du message SETUP original. La référence d'appel qui sera incluse dans le message ADD PARTY est identique à la référence d'appel incluse dans le message SETUP envoyé à la première feuille de la connexion et ce également pour toutes les futures feuilles de la connexion[5].

Un SIG-ADD_PARTY.ind est généré suite à la réception du message ADD PARTY au nœud A et indique une demande d'ajout d'une feuille. Le contrôle d'appel du nœud vérifie alors, tout comme pour un message SETUP, qu'il est possible de joindre le point d'accès au réseau du TE C (qui dans notre cas est également le nœud B). S'il est possible de joindre le nœud B, le message ADD PARTY est transporté à travers le réseau vers ce nœud. Remarquons cependant qu'il n'y a pas eu d'allocation d'un couple de valeurs (VPI, VCI) dans le nœud A : l'identifiant de connexion utilisé pour l'ajout de la feuille est totalement identique à celui utilisé pour la création de la connexion avec la première feuille et n'est pas inclus dans le message ADD PARTY[5].

Supposons que le message ADD PARTY soit arrivé au nœud B. Le TE C ne fait pas encore partie de la connexion point-à-multipoint en cours d'établissement; il n'existe donc pas de canal virtuel ouvert entre le TE C et le nœud B relatif à cette connexion. Dans ce cas, le nœud B va devoir ouvrir un canal virtuel jusqu'au TE C. La demande d'ajout d'une feuille ne sera alors plus transmise sous forme d'un message ADD PARTY mais sous la forme d'un message SETUP. Ce dernier contiendra : une nouvelle référence d'appel, un couple de valeurs (VPI, VCI) différent que celui assigné pour la connexion entre le nœud B et le TE B, la référence de point terminal reçue dans le message ADD PARTY, ainsi que les IE "classe de QoS" et "descripteur de trafic" délivrés dans le message SETUP reçu au nœud B et destiné à l'ajout de la feuille TE B. La procédure de connexion et l'échange de messages sont identiques à ce qui se déroule lors d'une connexion point-à-point entre le TE B et le nœud B de la Figure 3-8. Remarquons cependant que lorsque le nœud B reçoit un message d'acceptation de connexion CONNECT de la part du TE C, celui-ci est transmis à travers le réseau jusqu'au nœud B sous forme d'un message ADD PARTY ACK.

Les données reçues par les deux feuilles sont totalement identiques. De plus, les informations ne sont pas envoyées en multiples exemplaires dès la source, mais uniquement à partir du dernier nœud commun. Nous verrons dans les procédures exposées dans le chapitre consacré à la signalisation entre nœuds d'un réseau ce qu'est le concept de dernier nœud commun. Pour illustration, dans notre exemple, le dernier nœud commun est le nœud B : à partir de ce nœud, les chemins suivis pour arriver jusqu'aux feuilles sont différents. Ce n'est qu'à partir du dernier nœud commun que les données provenant du TE source sont dupliquées vers chacune des feuilles. Ceci permet une grosse économie en ressources du réseau vu que les données ne sont reproduites en plusieurs exemplaires qu'au dernier moment.

Nous avons dit que le message d'ajout d'une feuille ADD PARTY était converti dans le point d'accès au réseau du TE C en un message SETUP. Ceci n'est pas toujours le cas. En effet, si la nouvelle feuille est incluse dans le TE B avec lequel une connexion point-à-multipoint est déjà existante (cela peut être le cas si le TE B exécute deux programmes de réception des données en provenance de la racine ou si le

---

[5] La référence d'appel et le couple (VCI,VPI) sont identiques tant que l'on suit le chemin original qui avait été créé pour l'ajout de la première feuille. La section " Ajout d'une feuille à une connexion point-à-multipoint" du prochain chapitre, à la page 100, illustrera différent cas d'ajout de feuilles connectées à des points d'accès différents et pour lesquelles les chemins suivis à partir de la racine sont également dissemblables.

TE B est un commutateur IP recevant des données pour deux stations non ATM[6]) alors le message ADD PARTY ne serait pas converti en message SETUP mais passé tel quel au TE B. Celui-ci répondrait par un ADD PARTY ACKNOWLEDGE à son point d'accès

Lorsque le message ADD PARTY ACK arrive au nœud A après avoir traversé le réseau, le contrôle d'appel de ce nœud doit signaler l'acceptation de la connexion au TE A. Un SIG-ADD_PARTY.resp ayant comme paramètre la référence de point terminal de la feuille venant d'être ajoutée à la connexion génère l'envoi du message ADD PARTY ACK vers le TE A. Un SIG-ADD_PARTY.conf signale alors au contrôle d'appel du TE A la réception du message ADD PARTY ACK et la mise à disponibilité de la connexion avec la nouvelle feuille.

3.3.4.d) Procédure de redémarrage

La Figure 3-11 expose le flux de messages entre deux TE et leur point d'accès respectif au réseau pour une procédure de redémarrage.

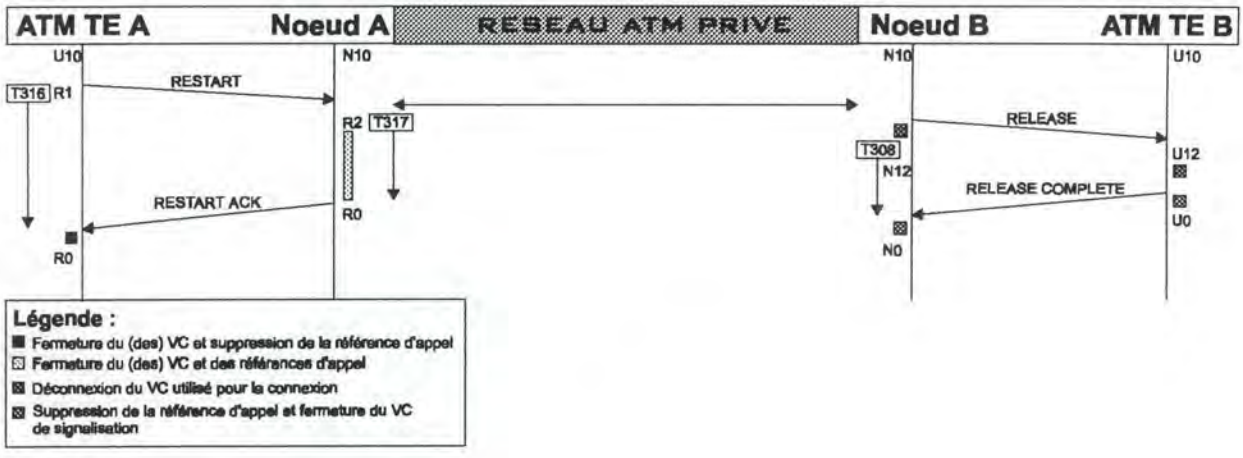


Figure 3-11 : flux de messages pour une procédure de redémarrage

La procédure de redémarrage peut être déclenchée :

- par le contrôle d'appel d'un nœud ou d'un TE si aucune réponse n'a été reçue après l'envoi d'un message RELEASE (i.e. expiration du timer T308 - voir annexe D);
- par le contrôle d'appel du TE A (ou B) dans le cas où le nœud A (ou B) ne répond plus aux messages de signalisation envoyés;
- par le contrôle d'appel du nœud A (ou B) dans le cas où le TE A (ou B) ne répond plus aux messages de signalisation envoyés.

Dans ces trois cas, la procédure de redémarrage se fait sur l'initiative du contrôle d'appel et non d'une application de plus haut niveau, i.e. l'utilisateur final n'a ni la capacité ni le droit de demander une procédure de redémarrage.

Supposons que dans notre cas le TE A souhaite fermer la connexion mais ne reçoit pas de message d'acquittement suite à l'envoi du message RELEASE. Supposons également que le contrôle d'appel souhaite demander le redémarrage de toutes les procédures de signalisation en cours. Dans ce cas, le

---

[6] Dans ce cas, le commutateur IP est le dernier 'host' ATM aux yeux de UNI.

contrôle d'appel génère un SIG-RESTART.req auprès de l'entité UNI où le paramètre Restart Indicator précise que tous les canaux virtuels contrôlés par le contrôle de protocole doivent être redémarrés. Ceci provoque l'envoi d'un message RESTART vers le nœud A.

La réception de ce message est signalée au contrôle d'appel du nœud A par un SIG-RESTART.ind. Cette primitive indique de plus quel est le VC à redémarrer. Dans notre cas, tout les VC sont à redémarrer. Le contrôle d'appel prend toutes les mesures internes nécessaires afin de retourner les VC concernés à l'état *null* et de rendre toutes les références d'appel et identifiants de connexion à nouveau disponibles pour toute procédure de connexion ultérieure. Rappelons que le message RESTART reçu à une signification locale uniquement; il ne sera donc pas transmis au delà du point d'accès. Toutefois, le nœud ayant fermé la ou les connexions existantes avec le TE A fermera également, par une procédure de déconnexion classique telle celle que nous avons vue à la section "Fermeture de connexion" page 58, les connexions qu'il a établis pour le compte du TE A.

Suite à ces actions, le contrôle d'appel du nœud A utilise un SIG-RESTART.resp générant l'envoi d'un message d'acquittement RESTART ACKNOWLEDGE vers le TE A. Suite à la réception de ce message (indiquée par un SIG-RESTART.conf), le TE A peut déconnecter tous les VC impliqués dans la procédure de redémarrage et rendre à nouveau libre pour toute procédure de connexion ultérieure les références d'appels et identifiants de connexion.

3.4 Conclusion

Nous avons vu dans ce chapitre ce qu'était un protocole de signalisation et quels étaient les services que ce protocole devait offrir. Nous avons également vu qu'ATM étant un protocole exclusivement orienté connexion, il était obligatoire d'avoir toujours 3 phases dans le cycle de vie d'une connexion entre TE : ouverture de connexion, transfert d'informations utilisateur puis fermeture de la connexion.

Dans ce chapitre, nous avons toujours parlé des flux d'informations existant entre un TE et son point d'accès au réseau. Nous avons toujours fait l'hypothèse du service offert par le réseau qui était de transporter les messages de signalisation vers le point d'accès au réseau du TE distant, mais nous ne savons pas encore comment fonctionne ce réseau, comment les messages arrivent vers leur destination, etc.

Le chapitre suivant présente le protocole PNNI de l'ATM Forum. Nous y verrons qu'il y a également des phases de signalisation entre chacun des nœuds du réseau se trouvant sur le chemin menant de la source à la destination de l'appel. Nous verrons également comment le chemin menant du point d'accès du TE appelant au point d'accès du TE appelé est trouvé.

UNI 3.1 est un protocole spécifié et approuvé par tous les membres de l'ATM Forum depuis près de deux ans et est déjà fortement implanté dans les réseaux ATM privés actuels. Nombreuses sont les sociétés qui ont implémenté UNI dans leurs produits ou dans leurs réseaux : 3Com Corp., Adaptec, Argile Networks, ATM Ltd., Bay Networks Inc., Cabletron systems Inc., Cascade Communications Corp., Cisco Systems Inc., Cross Comm Corp., Fore Systems, Newbridge Networks, Whitetree Network Technologies, ... Des sociétés telles Trillium Digital Systems Inc., Bellcore, Telenetworks et Harris & Jeffries ont développé et commercialisent des couches de signalisation UNI 3.0 ou UNI 3.1.

Le protocole UNI est actuellement en cours de spécification dans sa version 4.0. L'évolution de ce protocole a principalement pour but de suivre les évolutions dans le domaine d'ATM (par exemple : l'introduction de nouveaux types de trafic tel ABR). Ainsi, le protocole UNI 4.0 ajoute principalement à UNI 3.1 (liste non exhaustive) :

- le concept de *Leaf Initiated Join* (LIJ) : la procédure LIJ permet d'ajouter une feuille à une connexion point-à-multipoint sur demande de la feuille et non plus uniquement de la racine comme c'était le cas dans la version 3.1;

- la capacité de spécification précise du type de QoS que l'on demande : en particulier, il est possible de définir la QoS que l'on demande par l'intermédiaire de paramètres tels que CDV, CTD et CLR.
- le concept *d'Available Bit Rate* (ABR) : ABR est un nouveau type de trafic que l'on pourrait comparer à Ethernet. En effet, si un utilisateur précise le type de trafic ABR, il ne réserve aucune largeur de bande. Toute cellule émise par l'utilisateur et ne pouvant être placée sur le support physique par cause de manque de ressources est bufferisée et transmise dès que des ressources sont à nouveau disponibles;
- le concept de *Proxy Signaling* : ce concept permet à un utilisateur UNI, appelé le *Proxy Signaling Agent*, d'effectuer des procédures de signalisation pour un ou plusieurs utilisateurs qui ne supportent ou n'implémentent pas le protocole de signalisation;
- le concept d'ATM *anycast* : ce concept permet à un utilisateur de demander l'ouverture d'une connexion point-à-point avec n'importe quelle machine faisant partie d'un groupe identifié par une adresse de type *anycast*;

Le lecteur se référera à [UNI4.0-96] pour une liste exhaustive et une explication approfondie des nouveaux concepts et notions introduits dans la version 4.0 du protocole UNI.

4. Signalisation entre nœuds d'un réseau ATM privé

4.1 Introduction

Nous avons vu au chapitre 3 quelles étaient les procédures de signalisation à effectuer afin d'établir une connexion entre deux TE distants. Dans ce même chapitre, nous avons toujours parlé de l'échange de messages entre un TE et son point d'accès au réseau. Lorsque le TE appelé se trouvait connecté à un commutateur distant (i.e. un autre commutateur que celui sur lequel était connecté le TE appelant), nous avons fait l'hypothèse du service offert par le réseau ATM, c'est-à-dire le transport du message entre les deux commutateurs.

Ce chapitre a pour but d'étudier le service de transmission de messages de signalisation à travers le réseau. Nous étudierons plus particulièrement le protocole *Private Network to Network Interface*, encore appelé *Private Node to Network Interface* (PNNI). Ce protocole a comme rôle d'assurer les procédures de signalisation entre les commutateurs du réseau ATM, à partir du point d'accès au réseau du TE appelant jusqu'au point d'accès au réseau du TE appelé.

Nous voyons intuitivement qu'une autre dimension doit apparaître dans ce protocole, celle du choix d'un chemin à travers le réseau. En effet, alors que dans UNI 3.1 la question du choix d'un chemin pour un message ne se posait pas (le message allait toujours d'un TE à son point d'accès et vice versa), nous pouvons deviner que pour PNNI il en est tout autrement : il faut pouvoir à un moment donné choisir la route qui mènera du nœud appelant (i.e. le point d'accès du TE appelant) vers le nœud appelé. Le protocole PNNI peut donc être subdivisé en deux modules : un module de routage et un module de signalisation. La notion de routage semble paradoxale vu la nature orientée connexion du protocole ATM. Cependant il ne faut pas oublier qu'avant l'ouverture d'une connexion, il n'y a pas de connexion préalablement ouverte que suivrait une demande d'ouverture de connexion.

PNNI a la particularité d'être un réseau *hiérarchique*. Nous verrons dans une première section quelle est cette hiérarchie, ce qu'elle apporte et comment elle est construite à l'aide du module de routage. Nous verrons les protocoles particuliers utilisés pour la construction et le maintien de cette hiérarchie.

Une deuxième section présentera le module de signalisation, basé sur le protocole UNI 3.1. Nous verrons comment la signalisation tire profit de la représentation hiérarchique du réseau et, tout comme pour UNI 3.1, nous étudierons les états, messages et scénarios de procédures de signalisation de PNNI.

4.2 Vers PNNI

Rappelons que nous traitons dans ce mémoire des questions relatives à la signalisation dans les réseaux privés. Ces réseaux étant gérés par définition par des organismes privés, on peut alors supposer que le mécanisme de gestion interne (signalisation, acheminement des messages, ...) est propriétaire, i.e. propre à l'organisme qui les implémente. En effet, rien n'oblige ces organismes à implémenter des protocoles émis par des organismes de standardisation internationaux au sein de leurs réseaux.

Deux problèmes sont toutefois posés : l'interopérabilité des réseaux privés et le nombre très élevé d'informations devant être diffusées au sein d'un réseau afin que chaque nœud ait une vue précise de la topographie de celui-ci.

Traitons tout d'abord du problème de l'interopérabilité. La tendance actuelle et, à en croire les prévisions, pour les années à venir, est d'avoir un grand nombre de réseaux ATM privés interconnectés les uns aux autres. Il est donc impératif de définir un protocole permettant à ces réseaux de communiquer entre eux.

L'ATM Forum a perçu cette nécessité et a assigné le développement d'un protocole offrant cette interopérabilité à un groupe de travail[VIVID]. En attendant que la version définitive du protocole PNNI soit votée et acceptée par tous les membres de l'ATM Forum, celui-ci a émis une spécification pour le protocole *Interim Inter-switch Signaling Protocol* (IISP), également dénommé PNNI phase 0, initialement proposé par la société américaine Cisco.

Notons à nouveau que rien n'oblige les opérateurs de réseaux privés à implémenter des protocoles non-propriétaires tels que IISP et PNNI. Cependant, dans un but d'interopérabilité avec les autres réseaux privés, le protocole IISP ou PNNI devra être implémenté au moins aux frontières du réseau privé.

Le deuxième problème est celui de la diffusion d'information sur la topologie du réseau. Pour qu'un nœud puisse prendre des décisions de routage, il est nécessaire que celui-ci ait une vue précise de l'ensemble de la topologie du réseau (i.e. quels sont les nœuds présents, quelles sont leurs adresses, quels sont les TE qui y sont connectés et surtout, quels sont les liens - et leur capacité - connectant ces nœuds). Pour un réseau de très faible envergure (de 5 à 10 nœuds), la configuration manuelle ne pose pas trop de problèmes, mais lorsqu'il s'agit d'un réseau de 20 à 100 nœuds, voire plus, la configuration manuelle n'est pas du tout envisageable. Il faut donc qu'il existe un système permettant la diffusion automatique des données sur la topologie. Le protocole propriétaire SPANS, développé par la société américaine Fore Systems, est un protocole de signalisation permettant la diffusion automatique de ces données.

Nous proposons par la suite une brève présentation du protocole IISP ainsi que SPANS.

4.2.1 IISP (PNNI Phase 0)

Le texte qui suit est basé sur [IISP94].

L'Interim Inter-switch Signaling Protocol est principalement axé autour de la signalisation. C'est un protocole nettement moins complexe que PNNI phase 1 (étudié dans la suite de ce chapitre), basé principalement sur le protocole UNI 3.0/UNI 3.1 de l'ATM Forum. Vu le côté fortement symétrique des procédures de signalisation de ces protocoles (UNI 3.0 et UNI 3.1), IISP se base sur ceux-ci en assignant aux nœuds soit le rôle "utilisateur", soit le rôle "point d'accès au réseau" de ces protocoles. On peut alors définir deux types de lien particulier : les liens de type UNI et les liens de type IISP, tels que représentés à la Figure 4-1.

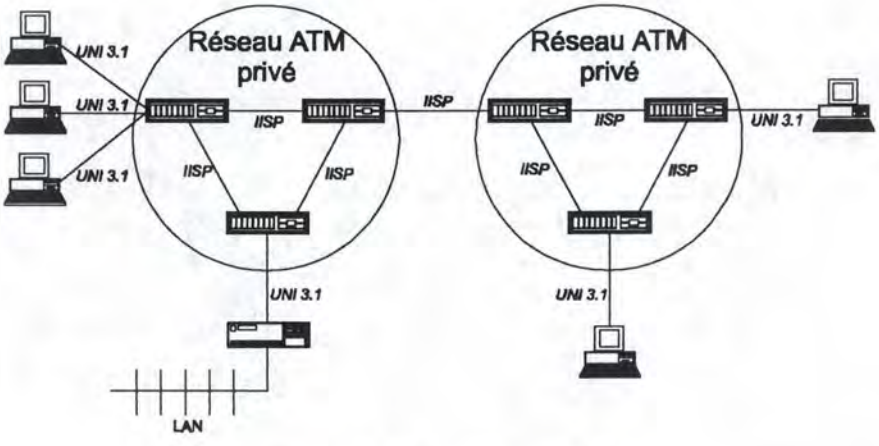


Figure 4-1 : liens IISP et UNI

Chaque nœud maintient localement une table de préfixes d'adresses. Lorsqu'un nœud reçoit un message d'ouverture de connexion sur un lien - qu'il soit UNI ou IISP -, il regarde dans sa table de préfixes et cherche le préfixe le plus long pouvant correspondre avec l'adresse contenue dans le message. Cette

table lui fournit alors le port de sortie correspondant au préfixe. Grâce à ce port, le nœud sait vers quel autre nœud il doit envoyer le message et utilise pour cette transmission les procédures des protocoles UNI 3.0/UNI 3.1.

L'avantage du protocole IISP est sa mise en œuvre très rapide. En effet il ne nécessite aucune modification des protocoles UNI 3.0/UNI 3.1 utilisés. Toute la "technologie" de routage est concentrée dans le contrôle d'appel.

L'inconvénient de ce protocole est la rigidité des tables de routage. Il n'y a aucune procédure automatique de mise à jour des tables et ceci doit donc être fait manuellement. IISP se prête donc bien aux réseaux privés à faible envergure (i.e. comportant un nombre très peu élevé de nœuds).

Par la suite, nous présenterons le protocole PNNI phase 1.

4.2.2 Autres solutions

D'autres solutions existent en dehors de IISP. Comme nous l'avons dit, chaque constructeur peut décider d'implémenter un protocole propriétaire. La société américaine FORE Systems par exemple a mis au point un protocole particulier s'appelant SPANS NNI, pour *Simple Protocol for ATM Network Signaling at the Network-to-Network Interface* [FORE95].

Nous ne détaillerons pas les mécanismes de ce protocole propriétaire. Notons cependant les avantages qu'il propose par rapport à IISP :

- un mécanisme de découverte de la topologie du réseau : chaque nœud découvre l'existence de liens le reliant à d'autres nœuds et publie cette informations. Tout nœud publie également les adresses des TE qui y sont connectés. L'avantage de ce mécanisme est de ne pas avoir à configurer manuellement les tables de routages comme cela se faisait dans IISP.
- un mécanisme de réallocation des VC : un VC transporté dans un VP particulier peut être assigné à un autre VP, pour peu que ceux-ci transportent les données dans la même direction. La demande de réallocation est émise par le nœud source (le point d'accès du TE appelant) vers le nœud terminal (le point d'accès du TE appelé). Ce mécanisme peut être enclenché suite à des baisses de performances sur le VP original ou en prévoyance de la fermeture de ce VP.
- un mécanisme de changement dans les ressources : les caractéristiques d'un VP reliant deux nœuds entre eux peut-être changée sur simple demande. Les caractéristiques d'un VP sont par exemple le nombre total de VC qu'il peut supporter. Si un nœud demande de modifier les caractéristiques d'un VP le reliant à un autre nœud et que cette modification entraîne des conséquences négatives pour un VC transporté dans ce VP, celui-ci pourra être rerouté via le mécanisme de réallocation des VC.

4.3 PNNI Phase 1

Cette section a été rédigée sur base de [PNNI94].

4.3.1 Un réseau hiérarchique

Comme nous l'avons dit dans l'introduction de ce chapitre PNNI a la particularité d'être basé sur une représentation hiérarchique des réseaux ATM. Clarifions ce terme et basons nous sur un exemple.

Considérons un réseau privé ATM constitué de 21 nœuds ou commutateurs, tel qu'illustré à la Figure 4-2.

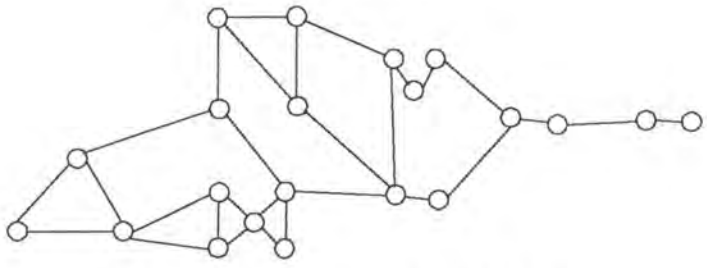


Figure 4-2 : un réseau privé ATM

Nous supposerons par la suite que le réseau exposé dans cette figure est géré par le même organisme, celui-ci ayant alors tout pouvoir d'attribution des adresses des nœuds et des TE qui y sont connectés (nous verrons que ceci a son importance pour la construction de la hiérarchie). Si toutefois l'ensemble des nœuds de la Figure 4-2 appartenaient à deux réseaux privés distincts, ceci ne changerait pas la logique utilisée pour la construction de la hiérarchie.

Dans un réseau non hiérarchique, chaque nœud a une vue plate du réseau : ceci veut dire que tout nœud de la Figure 4-2 doit avoir connaissance de tous les nœuds composants le réseau, de tous les liens les interconnectant ainsi que tous les TE connectés à l'ensemble des nœuds du réseau. En effet, lorsque le nœud source (i.e. le point d'accès au réseau du TE appelant) reçoit un message de type SETUP, il doit, à partir de l'adresse du TE appelé, trouver à quel nœud ce TE est attaché et surtout le chemin qui doit être suivi afin d'arriver jusqu'à ce nœud terminal.

Bien que cette solution soit envisageable pour un petit réseau ATM ne comportant que peu de nœuds, elle n'est pas envisageable pour des réseaux de taille plus importante, voire même planétaire. Le temps nécessaire pour parcourir les tables de routage à chaque nœud traversé entraînerait des performances médiocres.

Dans un réseau hiérarchique, les nœuds sont regroupés en un ensemble de groupes. Chaque groupe peut être représenté par un nœud « logique » qui est une représentation sous forme agrégée de l'ensemble des nœuds et des liens entre ceux-ci formant un groupe considéré. L'ensemble de tous les nœuds logiques que nous venons de créer - constituant un niveau de hiérarchie directement supérieur au niveau physique - peut être à nouveau considéré comme un ensemble de groupes. Et à nouveau, chacun de ces groupes peut être assimilé à un nouveau nœud logique qui représentera ce groupe à un niveau de hiérarchie supérieur.

Le but recherché par cette méthode est d'éviter que chaque nœud ait à retenir des informations précises à propos de l'ensemble du réseau, comme c'est le cas pour les réseaux non hiérarchiques. Avec la construction de la hiérarchie, on veut que tout nœud soit très bien informé sur tous les nœuds résidant dans son groupe et qu'il ait une vue de plus en plus agrégée du reste du réseau.

La section suivante clarifiera la construction de la hiérarchie. Nous construirons progressivement un réseau hiérarchique sur base de la Figure 4-2. Nous montrerons pour terminer la nouvelle vue du réseau qu'un nœud quelconque aura. L'avantage d'une construction hiérarchique d'un réseau ATM sera alors évident.

Notons toutefois que l'idée d'un réseau hiérarchique n'est pas une innovation propre au protocole PNNI. L'idée est basée sur des travaux de recherche et développement faits à l'IETF et portant le nom de Nimrod. Le but recherché par le projet Nimrod est de constituer une nouvelle structure d'adressage pour les réseaux TCP/IP et ce par l'intermédiaire d'une hiérarchie dans la structure des adresses [VIVID].

4.3.1.a) Construction de la hiérarchie

i - Premier niveau

Comme nous l'avons introduit dans la section précédente, nous allons commencer par regrouper les nœuds du réseau de la Figure 4-2 en groupes. Dans le protocole PNNI, ces groupes sont appelés ***Peer Group*** (PG).

Tout nœud possède une adresse ATM privée sur 20 octets, telle que définie dans le premier chapitre. La construction des groupes s'articule autour du concept de préfixes sur les adresses ATM. Par définition, un préfixe sur une adresse ATM est constitué, dans la construction des PG, des premiers p bits de cette adresse, la valeur de p variant de 0 à 104 (on ne considère donc que les 13 premiers octets de haut niveau de l'adresse ATM et pas les champs ESI et SEL de celle-ci). Un PG est constitué par des nœuds ayant le plus long préfixe commun sur leur adresse. Ajoutons également que les nœuds choisis pour constituer un PG doivent être obligatoirement connecté entre eux par un lien physique. Il serait inutile de regrouper des nœuds ne sachant pas communiquer entre eux.

Tout PG a un niveau définissant le niveau de hiérarchie auquel il se trouve. Ce niveau correspond simplement à la longueur du préfixe sur adresses qui a été choisi. Plus un PG se trouve haut dans la hiérarchie plus son préfixe est court et inversement. On peut alors identifier un PG par un couple (niveau, préfixe) constituant le PG Identifier (PGID).

Considérons à nouveau le réseau de la Figure 4-2 et regroupons les nœuds en PG.

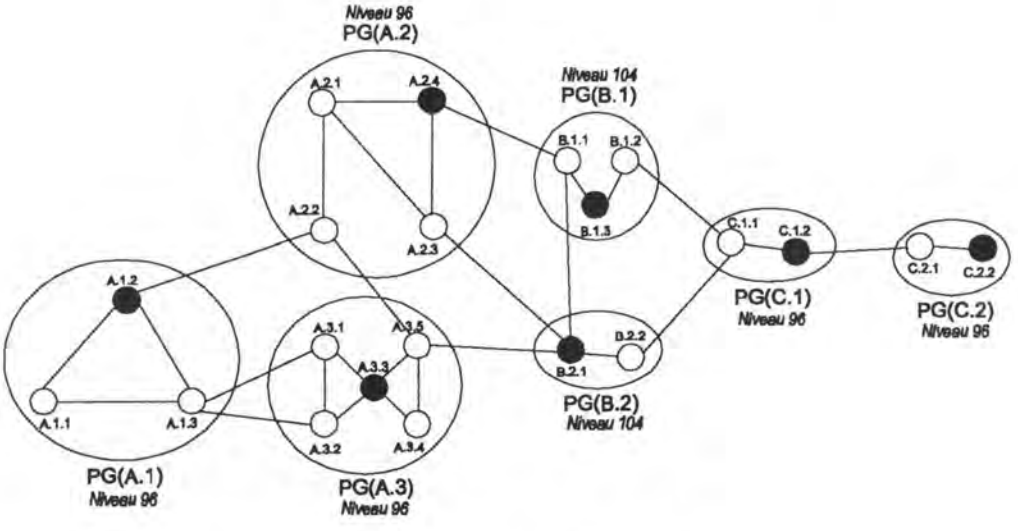


Figure 4-3 : construction des PG

Afin de ne pas surcharger cette figure, les adresses des nœuds sont représentés par des valeurs A.1.1, A.1.2, etc. Ceci ne modifie en rien la logique appliquée.

Dans la Figure 4-3, les nœuds A.1.1, A.1.2 et A.1.3 ont un préfixe commun A.1, de longueur hypothétique égale à 96 (donc les 12 premiers octets des adresses). De même les nœuds A.2.1, A.2.2, A.2.3 et A.2.4 ont été regroupés dans le PG de préfixe commun A.2, de niveau 96. Il en va de même pour la constitution des autres PG. Remarquons que les niveaux des PG constitués ne sont pas tous identiques. Ainsi, pour les nœuds B.2.1 et B.2.2, le plus long préfixe commun B.2 a une longueur de 104 bits.

Nous remarquons par la construction de ces PG qu'il doit y avoir correspondance entre la topologie du réseau et la distribution des adresses. Il en va donc de la responsabilité de l'administrateur du réseau

d'assigner les adresses de manière à permettre la construction de PG. Nous avions fait cette hypothèse en début de section.

Nous nous trouvons toujours pour l'instant au niveau le plus bas de la hiérarchie. Nous avons jusqu'ici fait référence au concept de nœud. Par la suite, nous utiliserons également le terme "nœud logique". Un **nœud logique** est soit un nœud physique (se trouvant donc au niveau le plus bas de la hiérarchie), soit un nœud représentant un PG à tout niveau de hiérarchie supérieur. Nous verrons par la suite ce concept de nœud à un niveau autre que le niveau physique. Un type particulier de nœud est le **nœud frontière** : il s'agit d'un nœud ayant un ou plusieurs liens sortant de son PG (par exemple : A.1.2, A.1.3, A.2.2, ...).

Nous avons également fait référence au concept de lien entre les nœuds. Par la suite, nous utiliserons également le terme « lien logique ». Un **lien logique** est soit un lien physique (reliant deux nœuds physiques), soit un lien reliant deux nœuds logiques (à un niveau différent que le plus bas niveau de la hiérarchie, i.e. le niveau physique). De plus, nous pouvons dès à présent différencier deux types particuliers de lien : dans la Figure 4-3 les liens reliant des nœuds se trouvant dans le même PG sont appelé **liens horizontaux** et les liens reliant deux nœuds se trouvant dans des PG différents sont appelés **liens extérieurs**.

Nous avons dit à la section 4.3.1 que le but de la représentation hiérarchique d'un réseau est qu'au sein d'un même groupe, tous les nœuds aient une information la plus précise possible sur l'ensemble des nœuds et des liens inclus dans ce groupe, puis une représentation de plus en plus agrégée de l'ensemble du réseau et donc des autres PG. Pour ce faire, il existe trois procédures particulières dont le but est précisément l'échange d'informations au sein d'un PG et permettant la récolte d'informations précises quant aux nœuds et liens contenus dans le PG : il s'agit de la machine *Hello*, de la synchronisation de base de données sur la topologie et de la procédure d'inondation.

Présentons ces trois procédures.

A chaque lien (logique) que possède un nœud (logique), on associe une machine à états finis particulière appelée machine *Hello* (exposée à la page 77). Dès qu'un lien devient actif, la machine Hello démarre et continue à tourner jusqu'à ce que ce lien ne soit plus actif. Grâce à cette procédure, deux nœuds logiques voisins (directement connectés entre eux par un lien logique) se communiquent leur adresse ATM, leur PGID et un identifiant du port attaché au lien logique. Le but principal de cette procédure est donc la découverte des nœuds voisins.

Ces informations circulent dans des structures particulières appelées paquets Hello. Grâce à ce mécanisme, tout nœud récolte des informations concernant ses voisins directs et apprend entre autres si les nœuds voisins font partie du même PG que lui. Les nœuds découvrant qu'ils ne font pas partie du même PG n'exécutent pas les procédures de synchronisation de base de données et d'inondation exposés par la suite.

Notons qu'un VC particulier est réservé à l'échange de toutes les informations de routage tels les paquets Hello. Il est identifié par le couple (VPI=0, VCI=18) et appelé *Routing Control Channel* (RCC).

Nous avons vu que les nœuds voisins s'échangeaient entre autres leur PG ID. Des nœuds voisins se rendant compte qu'ils appartiennent au même PG vont synchroniser leur base de données sur la topologie.

Tout nœud dispose d'une base de données sur le topologie du réseau. Cette base de données contient des informations de 3 types :

- informations quant aux TE qui sont connectés aux nœuds (leur adresse);
- informations sur le nœud lui-même (typiquement : son adresse ATM);
- informations sur les liens connectés aux nœuds (leur description en terme de capacité) et sur les nœuds logiques (si, comme nous le verrons plus tard, un nœud logique représente un PG

à un niveau supérieur de la hiérarchie, cette information est une représentation agrégée du PG qu'il représente).

Ces informations sont contenues dans des structures appelées PTSE (*PNNI Topology State Element*). Initialement (i.e. au démarrage du nœud), cette base de données ne contient que des informations connues directement par le nœud : son adresse ATM, la description des liens qui sont connectés à ce nœud et l'ensemble des TE qui y sont connectés (pour peu qu'il y en ait).

Synchroniser les bases de données signifie que les nœuds voisins appartenant au même PG vont coordonner le contenu de leur base de données afin que celles-ci soient identiques dans toutes les paires de nœuds voisins (les bases de données seront donc constituées des mêmes PTSE). Le mécanisme de synchronisation des bases de données sur la topologie est exposé à la page 79.

Le troisième mécanisme utilisé est celui de l'inondation. Alors que le rôle de la procédure de synchronisation de la base de données sur la topologie était d'assurer que toutes les paires de nœuds voisins aient une même base de données concernant la topologie, le rôle du mécanisme d'inondation est d'assurer, par l'intermédiaire d'échanges de paquets particuliers, que *tous* les nœuds logiques constituant un PG ont la même base de données sur la topologie. Ce mécanisme est exposé à la page 80.

Nous avons donc constitué à la Figure 4-3 un ensemble de PG. Dans chacun de ces PG, tous les nœuds se connaissent l'un l'autre et ont toutes les informations permettant de router une procédure de connexion à l'intérieur même de leur PG. Mais tous les nœuds de ces PG ne connaissent que le "monde intérieur" et n'ont donc aucune connaissance sur le "monde extérieur" (tout ce qui se passe à l'extérieur de leur PG). Il nous faut donc passer à un niveau de hiérarchie supérieur.

ii - Ajout d'un niveau

Dans chaque PG va avoir lieu une procédure d'élection. Le but est de choisir un nœud qui représentera le PG au niveau de hiérarchie directement supérieur. Ce nœud logique particulier est appelé le ***Peer Group Leader*** (PGL). Un petit sous-protocole particulier a pour but de gérer ce processus d'élection. Nous ne le présenterons pas ici car il n'apporterait aucune information importante. Notons simplement qu'à chaque nœud est attribué (à la configuration) un « indice préférentiel ». Le nœud du PG ayant l'indice le plus élevé est élu PGL de ce PG.

A la Figure 4-3, les PGL de chaque PG ont été noircis. Le PGL n'a pas d'autre rôle que celui de représenter son PG à un niveau supérieur de la hiérarchie. Remarquons qu'un PG ne doit pas avoir de PGL afin d'effectuer un routage interne au PG. De même, si le réseau privé n'est constitué que d'un seul PG, il est inutile que ce PG ait un PGL.

Dans l'exemple de la Figure 4-3 nous avons divisé le réseau en plus d'un PG. Chaque PG va donc élire un PGL. Nous aurons donc un « représentant » pour chaque PG à un niveau hiérarchique supérieur.

Considérons à présent la Figure 4-4.

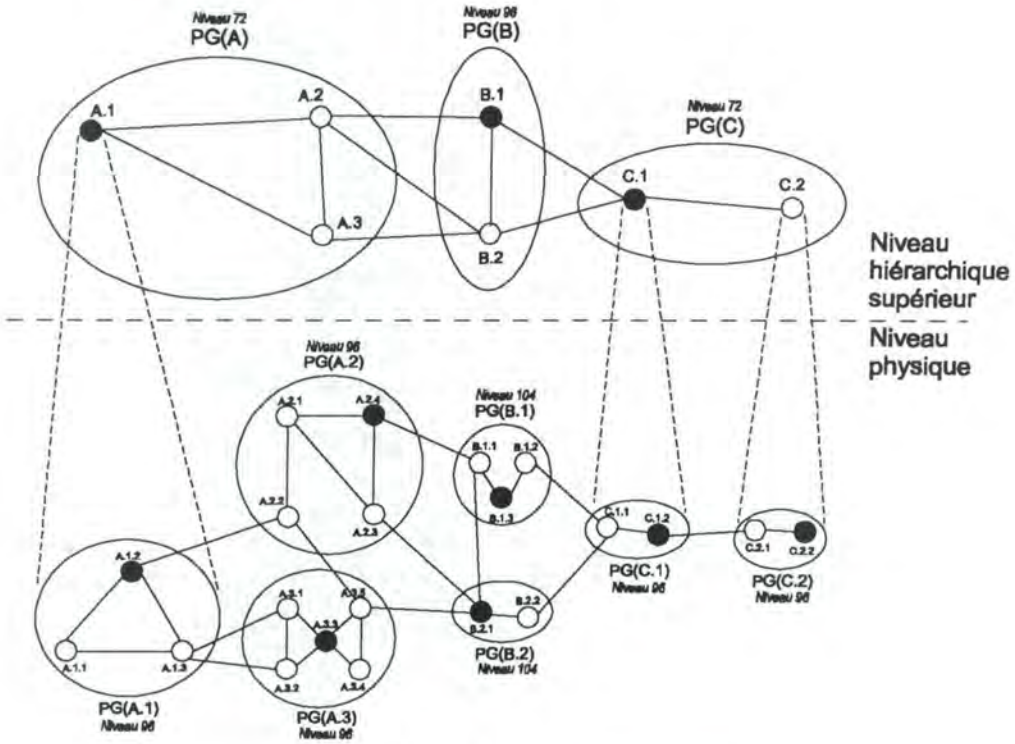


Figure 4-4 : deux niveaux hiérarchiques

Prenons plus particulièrement le PG A.1. Ce PG va être représenté au niveau hiérarchique directement supérieur par son PGL, ici le nœud A.1.2. Ce PGL est donc un nœud logique au niveau supérieur de la hiérarchie. Le nœud logique A.1 est ce PGL A.1.2. Le nœud physique d'adresse ATM A.1.2 assume donc jusqu'ici un double rôle : celui d'un nœud physique appartenant au PG A.1 et celui de représentant du PG A.1 au niveau supérieur.

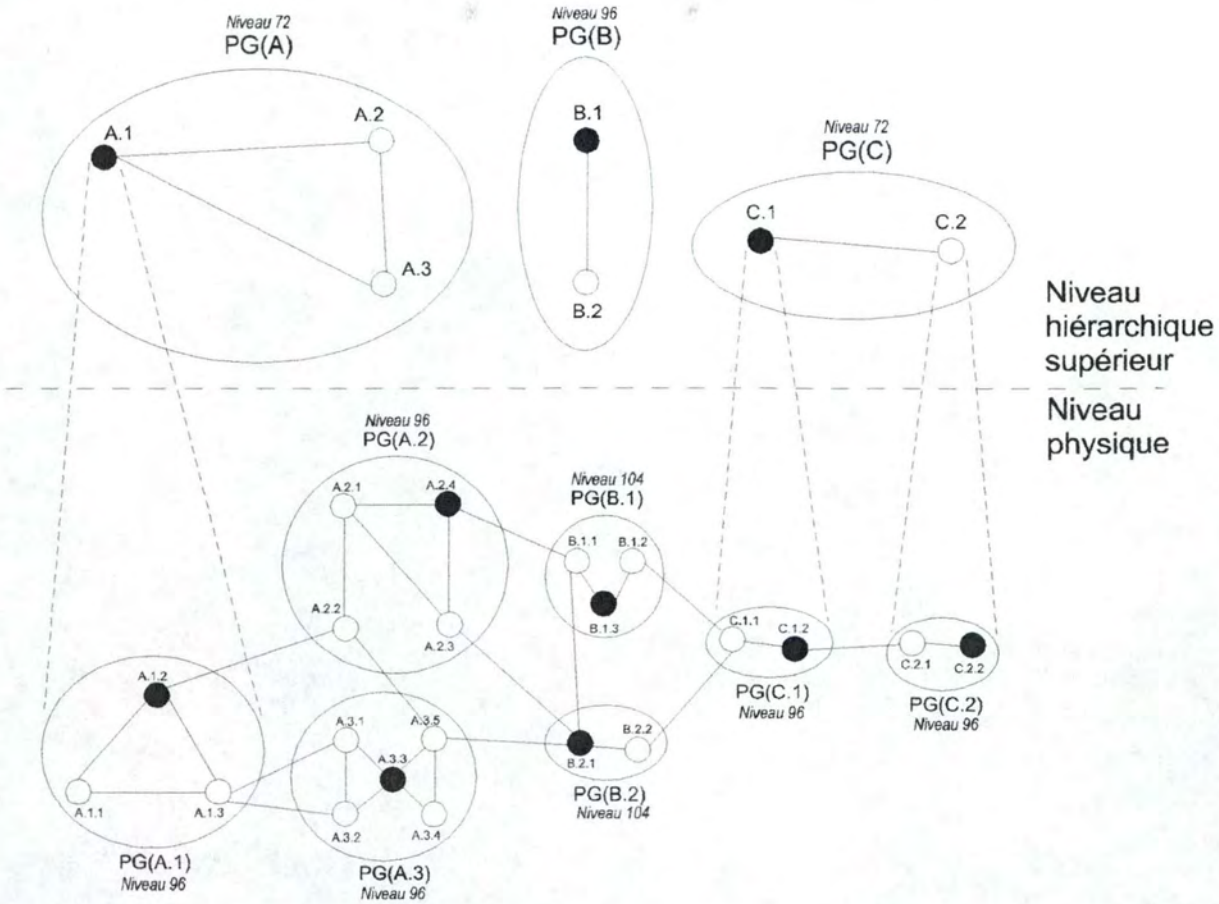
On obtient donc à nouveau un ensemble de nœuds que l'on va rassembler en groupes selon le même procédé que celui utilisé au niveau de hiérarchie le plus bas, c'est-à-dire par préfixe commun sur les adresses. Ainsi, les nœuds logiques A.1, A.2 et A.3 ont le préfixe commun A, de longueur hypothétique commune égale à 72. Nous venons donc de constituer un nouveau PG A, de niveau 72. Le PG A est appelé **PG parent** des PG A.1, A.2 et A.3, ceux-ci étant appelés les **PG enfants** du PG A.

Nous savons que dans le PG A.1, le PGL A.1.2 a toutes les informations concernant la topologie interne du PG A.1 (par le biais de la machine Hello, de la synchronisation des bases de données sur la topologie et par la procédure d'inondation) : description de tous les liens, adresses de tous les nœuds contenu dans le PG A.1 ainsi que toutes les données d'accessibilité. Le rôle du nœud A.1 (et donc du PGL A.1.2) va être de distribuer ces données aux autres nœuds logiques appartenant au même PG A. Plutôt que de distribuer telles quelles ces informations, celles-ci vont être résumées. Ainsi, pour les données d'accessibilité, le nœud logique A.1 va tenter de résumer les adresses de tous les TE se trouvant dans le PG A.1. Un exemple de résumé d'adresses est donné à la page 80. De même, les données concernant la topologie (liens et nœuds) vont être représentées dans une structure particulière agrégée.

Un nœud logique tel que A.1 aura également pour rôle de distribuer l'information qu'il aura reçue des autres nœuds logiques A.2 et A.3 (pendant les phases de synchronisation de base de données et d'inondation se déroulant à ce niveau) à tous les nœuds du PG qu'il représente.

Le problème restant toutefois posé est celui de la communication entre nœuds logiques du niveau hiérarchique que nous venons de constituer. Si au niveau le plus bas, c'est-à-dire physique, la question ne se posait pas vu l'existence de liens physiques entre nœuds appartenant à un même PG, il n'existe ici

- Figure 4-4, page 72 : liens logiques inexistants entre les PG(A), PG(B) et PG(C)



aucun lien physique entre les nœuds logiques[7] A.1, A.2, etc. Il va alors être nécessaire d'établir un SVC entre les nœuds logiques A.1, A.2, etc., afin de permettre l'exécution des procédures Hello, synchronisation de base de données et inondation.

Voici la procédure qui va être utilisée à cet effet :

1. nous savons que la machine Hello tourne entre tous les nœuds logiques appartenant au niveau de hiérarchie le plus bas. Cette procédure tourne également entre deux nœuds n'appartenant pas au même PG. Ces nœuds particuliers avaient été dénommés nœuds frontière. Dans la Figure 4-4, les nœuds A.1.2 et A.2.2 sont des nœuds frontière n'appartenant pas au même PG. L'information que donnera le nœud A.1.2 au nœud A.2.2 (par exemple) couvrira entre autres l'adresse ATM du nœud logique A.1 (donc du PGL A.1.2) représentant le PG A.1 au niveau de hiérarchie supérieur. De même, A.2.2 donnera à A.1.2 l'adresse ATM du nœud logique A.2 (donc du PGL A.2.4) représentant son PG au niveau de hiérarchie supérieur. A partir de ces informations, A.1.2 et A.2.2 vont tenter de déterminer s'ils ont un PG en commun. Dans le cas de la Figure 4-4, A.1.2 et A.2.2 ont le PG A en commun.
2. A.1.2 a découvert qu'il avait un lien avec le nœud A.2.2, représenté par le nœud logique A.2 dans le PG A commun. A.1.2 va avertir l'ensemble des nœuds de son PG de l'existence d'un lien particulier appelé **uplink** le connectant au nœud logique A.2 appelé dans ce cas **upnode** (bien que le lien mène au nœud A.2.2, on désire, toujours dans un soucis d'agrégation de la topologie, résumer la connexion et dire qu'elle mène au nœud A.2). Il va utiliser à cet effet la procédure d'inondation. Le lien de type uplink indique que A.1.2 a un lien vers le nœud logique A.2. De même, A.2.2 va avertir l'ensemble des nœuds de son PG qu'il a un uplink vers le nœud logique *A*.1 (qui est donc considéré comme *upnode*).
3. l'information à propos des uplinks découverts étant envoyée à tous les nœuds des PG concernés, les PGL seront donc finalement avertis de l'existence de ces uplinks. Dans la Figure 4-4, le nœud logique A.1 - qui comme nous le savons n'est rien d'autre que le PGL A.1.2 - sait donc qu'il existe un nœud logique A.2 se trouvant dans le même PG A que lui. De plus, A.1 sait qu'il doit exister un lien entre lui et le nœud logique A.2 vu la publication de l'uplink. Le nœud logique A.1 va donc demander l'ouverture d'un SVC avec le nœud logique A.2. Afin d'établir ce SVC, les procédures du module de signalisation du protocole PNNI seront utilisées.

Reprenons la Figure 4-4. Dans cet exemple, le nœud logique A.1 a appris par le mécanisme que nous venons d'exposer qu'il existait deux nœuds logiques A.2 et A.3 se trouvant dans le même PG A que lui et, par la publication des uplinks, a découvert qu'il y avait connectivité avec ces deux nœuds. A.1 a donc demandé l'ouverture d'un SVC avec chacun de ces nœuds. Il en va de même pour A.2 et A.3 ainsi que pour la construction des PG B et C. Notons toutefois le cas particulier de la connectivité entre les nœuds logiques A.1 et A.3. Nous voyons en effet qu'au niveau hiérarchique le plus bas, il existe deux liens physiques allant du PG A.1 vers le PG A.3. Cependant, au niveau de hiérarchie supérieur (i.e. dans le PG A), il n'y a qu'un seul lien logique entre les nœuds logiques A.1 et A.3. Comme nous l'avons dit, le but de la construction hiérarchique est principalement de pouvoir agréger les données concernant la topologie et l'accessibilité au fur et à mesure que l'on monte dans les niveaux hiérarchiques. Les deux liens allant de A.1.3 vers A.3.1 et A.3.2 ont été agrégés et sont représentés par un seul lien logique entre A.1 et A.3.

---

[7] Notons que ceci pourrait arriver : considérant l'exemple de la Figure 4-4, si les PGL des PG A.1 et A.2 sont les nœuds A.1.2 et A.2.2, on voit qu'il existe alors un lien logique entre ces PGL et donc entre les nœuds logiques de niveau supérieur qu'ils représentent. Toutefois, vu que l'on s'adresse à des niveaux et donc à des rôles différents, ceci n'altère en rien la procédure exposée par la suite.

Les SVC ainsi créés vont être utilisés comme RCC. Donc, tout comme pour le niveau hiérarchique le plus bas, les informations agrégées distribuées entre les nœuds logiques du PG A vont être distribuées à l'aide d'une machine Hello, d'une synchronisation des bases de données et d'une procédure d'inondation. De même une procédure d'élection d'un PGL au sein des PG A, B et C va avoir lieu.

iii - Dernier niveau

Comme nous pouvons le constater à la Figure 4-4, la construction de la hiérarchie n'est pas complète. Il manque principalement la connectivité entre les nouveaux PG A, B et C que nous avons créés.

Comme nous l'avons dit pour terminer la section précédente, un processus d'élection de PGL va à nouveau être amorcé dans chaque nouveau PG. Le même processus de construction que celui appliqué lors de la section précédente va être utilisé.

On ne construit plus de niveau de hiérarchie lorsque tout le réseau est représenté par un seul PG, comme c'est la cas à la Figure 4-5.

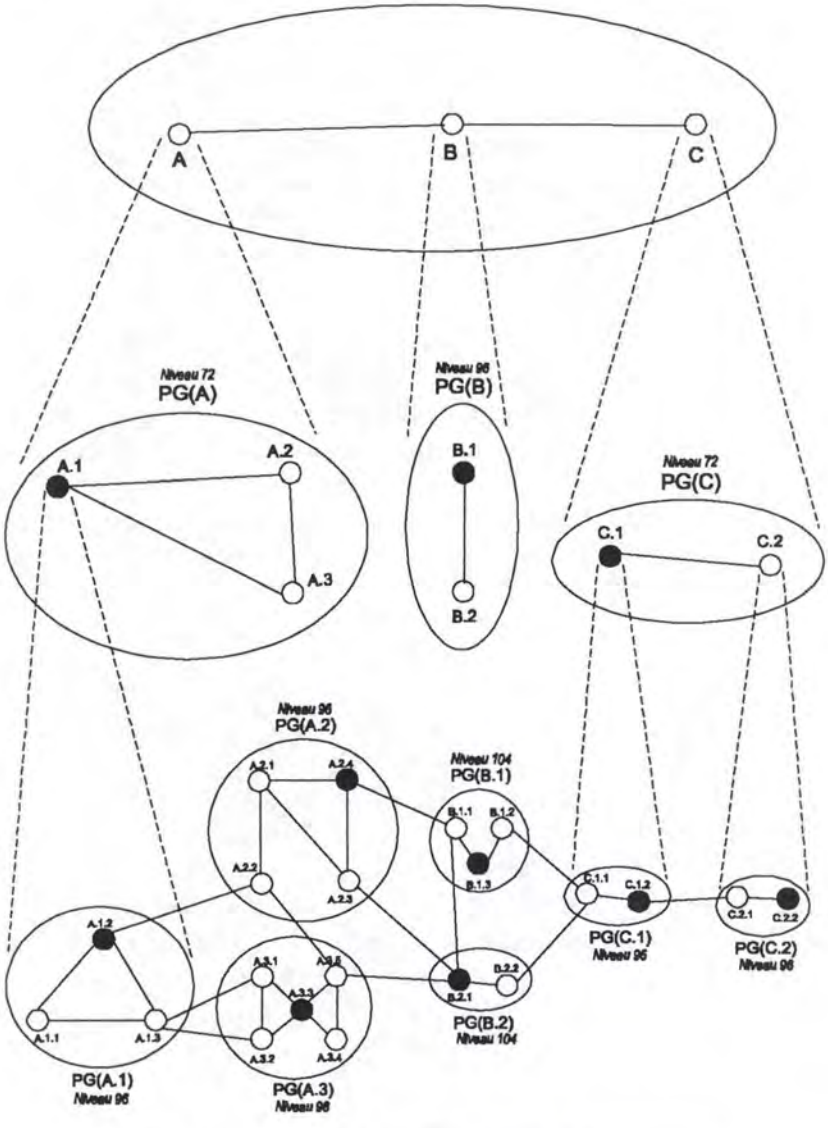


Figure 4-5 : ajout du dernier niveau de la hiérarchie

Notons cependant deux différences avec le phase d'ajout d'un niveau de la section précédente :

1. il n'y a plus d'élection de PGL dans le dernier niveau de hiérarchie. En effet ce PGL n'aurait aucun rôle puisqu'il ne devrait plus représenter de PG à un niveau de hiérarchie supérieur.
2. les liens logiques entre les nœuds logiques A, B et C de la Figure 4-5 ne peuvent être tirés de l'exécution d'une machine Hello au niveau des PG A.1, A.2, A.3, B.1, etc... Ils sont tirés par contre des uplinks découverts au plus bas niveau de la hiérarchie.

Remarquons que le niveau d'un PG est compris entre 0 et 104. Le fait de pouvoir définir un préfixe de longueur 0 permet la création d'un PG pour lequel il n'est pas possible de trouver un préfixe commun aux nœuds logiques le constituant. Ainsi, si à la Figure 4-5 il n'est pas possible de trouver un préfixe commun aux nœuds logiques A, B et C, cela ne pose aucun problème. Nous voyons alors que l'interconnexion de réseaux ATM privés gérés par des organismes différents ne pose aucune difficulté, même s'il n'est pas possible de trouver de préfixe commun entre les nœuds logiques représentant ces réseaux au niveau hiérarchique le plus élevé.

iv - Hiérarchie perçue par un nœud physique

Nous avons mis en avant à la section 4.3.1 que le but de la hiérarchisation du réseau était d'obtenir une vue de plus en plus agrégée du réseau afin qu'un nœud physique n'ait pas à connaître tous les nœuds le composant, les liens interconnectant ceux-ci et l'ensemble des TE qui y étaient connectés. Sur base de la construction de la hiérarchie telle que nous l'avons exposée, tentons de voir quelle est la vue qu'un nœud quelconque a de l'ensemble du réseau. Prenons par exemple le nœud A.1.1.

Nous savons qu'au sein du PG A.1 dans lequel se trouve le nœud physique A.1.1 un échange précis sur la topologie interne du PG a été effectué. A.1.1 connaît donc précisément la topologie interne de son PG, ce que nous avons représenté à la Figure 4-6.

L'échange d'informations précises de topologie ne s'est pas étendu à l'extérieur du PG A.1. En effet, la machine Hello, la synchronisation des bases de données sur la topologie et la procédure d'inondation ne s'étendaient pas au-delà des frontières du PG. Par contre, nous savons qu'un tel type d'échange de données a eu lieu dans le PG A de la Figure 4-4. Les données qui étaient échangées étaient des résumés sur la topologie et l'accessibilité des PG représentés par les nœuds logiques A.1, A.2 et A.3. Le rôle du nœud logique A.1 était en autre de répercuter ces données à tous les nœuds physiques constituant le PG A.1. Le nœud physique A.1.1 connaît donc l'existence des nœuds logiques A.1 (qui représente son PG), A.2 et A.3 ainsi que les liens logiques reliant son PG à ces deux nœuds (ces uplinks, rappelons nous, avaient été publiés dans l'ensemble du PG A.1 par A.1.2 et A.1.3) et les liens logiques existant au sein du PG A (ceux-ci avaient été publiés par le nœud logique A.1 - donc le PGL A.1.2 - au sein du PG A.1).

Toujours sur base de la Figure 4-4, nous savons qu'il n'y a pas de connaissance des PG B.1, B.2, C.1 et C.2 à ce niveau. Par contre, le nœud logique A de la Figure 4-5 connaît les nœuds B et C ainsi que les liens logiques les interconnectant. Tout comme pour le nœud logique A.1, le nœud A va diffuser au sein du PG A qu'il représente les données agrégées qui auront été échangées dans le PG résidant au niveau le plus élevé de la hiérarchie. Par ce fait, les nœuds logiques A.1, A.2 et A.3 sont au courant de l'existence des nœuds logiques B et C et des liens logiques les interconnectant. Ces informations sont distribuées entre tous les nœuds logiques du PG A et, par l'intermédiaire du nœud logique A.1, seront distribuées au sein du PG A.1. Par la procédure d'inondation, le nœud physique A.1.1 sera finalement au courant de l'existence des nœuds logique B et C et des liens logiques les interconnectant.

La Figure 4-6 illustre finalement la vue du réseau qu'a le nœud physique A.1.1. Il est évident que ce type de représentation est nettement plus léger que celui de la Figure 4-2. Le nœud A.1.1 n'a plus à mémoriser une topologie complexe constituée de 22 nœuds, de 29 liens physiques (ainsi que leurs caractéristiques propres) et de l'ensemble des TE qui y sont connectés. En lieu et place, le nœud A.1.1

doit maintenir une connaissance sur 9 nœuds, 8 liens logiques et des données d'accessibilité fortement réduites vu le mécanisme de résumé d'adresses.

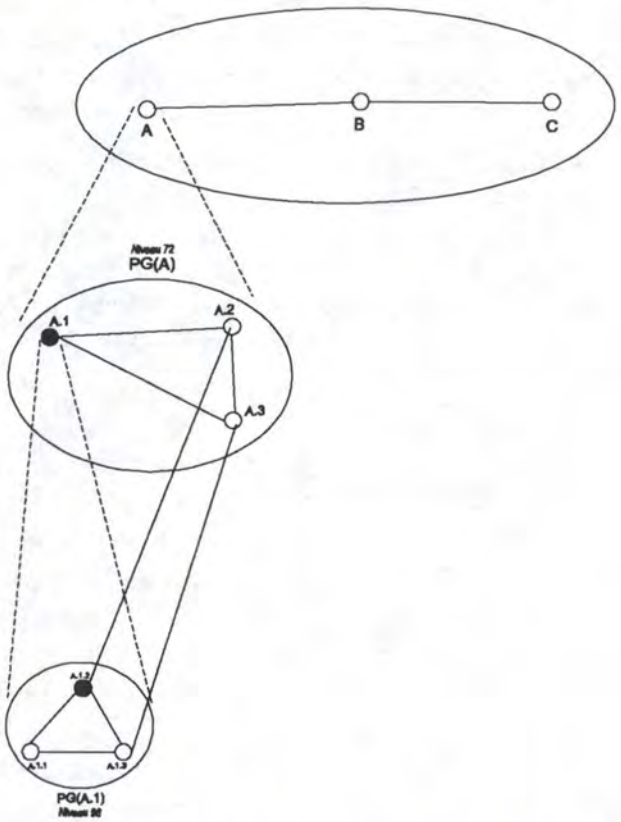


Figure 4-6 : vision globale du réseau par un nœud physique

4.3.1.b) Pour conclure

Alors qu'il était facile de s'imaginer comment un nœud tel que A.1.1 pouvait router une demande de connexion à travers le réseau quand il connaissait la topologie complète et précise de l'ensemble de celui-ci, on peut se demander comment ce nœud pourra effectuer le même travail avec une représentation hiérarchique du réseau tel qu'il la maintient dorénavant. Le module de signalisation de PNNI est très fortement basé sur le protocole UNI 3.1 de l'ATM Forum. Nous verrons dans la section consacrée à la signalisation les structures complémentaires qui ont été mises en place permettant de tirer profit de la construction hiérarchique dans le but de router les appels.

Avant de présenter cette section, nous présenterons dans une section consacrée au routage la machine Hello, le mécanisme de synchronisation de la topologie, la procédure d'inondation et le principe de résumé d'adresses.

Nous savons également de par le chapitre 1 que lors d'une demande d'ouverture de connexion, chaque nœud impliqué dans cette procédure doit vérifier s'il peut supporter les caractéristiques de trafic demandées par l'utilisateur. Cette fonction, appelée CAC, fait partie du module de routage du protocole PNNI. Elle sera présentée, ainsi que la fonction de routage, dans cette même section.

4.3.2 Module de routage

Ce mémoire étant consacré à la signalisation dans les réseaux ATM privés, nous ne nous attarderons pas sur une description exhaustive du module de routage. Par contre, nous présenterons de manière globale dans une première sous-section les procédures utilisées pour la construction de la hiérarchie exposées à la section 4.3.1a), à savoir la machine Hello, la procédure de synchronisation des bases de données sur la topologie, la procédure d'inondation ainsi que le principe de résumé d'adresses.

Une deuxième sous-section abordera les concepts de Call Admission Control et de sélection de route pour une procédure de connexion (le *path selection*).

4.3.2.a) Procédures utilisées pour la construction de la hiérarchie

i - Machine Hello

Nous devons différencier deux machines Hello différentes bien qu'ayant les mêmes fonctionnalités : la machine Hello tournant au niveau le plus bas de la hiérarchie et celle tournant entre nœuds logiques à tout autre niveau de la hiérarchie. Ces machines se différencient principalement par la structure de données échangée.

Une première sous-section présentera la machine Hello tournant au niveau le plus bas de la hiérarchie. Dans une deuxième sous-section, nous présenterons la machine Hello tournant à tous les autres niveaux de la hiérarchie.

I. Machine Hello au niveau le plus bas de la hiérarchie

La machine Hello a pour but de permettre à des nœuds voisins de découvrir leur existence réciproque et d'échanger des informations permettant principalement de détecter s'ils appartiennent à un même PG. Un nœud découvrant un lien physique le connectant à un autre nœud[8] démarre une machine Hello associée à ce lien et échange à travers celui-ci des informations avec le nœud distant. On aura donc autant de machines Hello en exécution dans un nœud qu'il y a de liens physiques connectant ce nœud à des nœuds voisins.

Comme nous l'avons dit à la section décrivant la construction d'une hiérarchie, la machine Hello est en exécution permanente jusqu'à ce que le lien reliant les deux nœuds voisins ne soit plus actif[8] (i.e. un des nœuds a été désactivé).

Echange de données

Une structure de données appelée paquet Hello est échangée à travers les RCC et transporte les éléments suivants :

- *Identifiant du port* : un numéro assigné par le nœud permettant d'identifier le port physique et le VC (i.e. le RCC) associé à cette structure de données.
- *Identifiant du nœud* : Chaque nœud logique, qu'il soit au plus bas niveau de la hiérarchie ou à un niveau plus élevé, a un identifiant autre que son adresse ATM. Cet identifiant (le **node ID**) est constitué de 22 octets. Pour un nœud résidant au niveau hiérarchique le plus bas, le

[8] Nous avons vu dans le chapitre consacré à UNI que le premier protocole qui était activé était le protocole ILMI. Ceci se passe également dans PNNI (mis à part le fait qu'il n'y a pas de procédure d'allocation d'adresse entre nœuds du réseau). L'indication de disponibilité du lien vient de ILMI. De même lorsque le lien n'est plus disponible.

premier octet de cet identifiant indique le niveau de son PG. Le deuxième octet de l'identifiant du nœud vaut toujours 160 (valeur fixe) et les 20 derniers octets sont complétés par l'adresse ATM du nœud.

- *Identifiant du nœud distant* : cette information n'est pas diffusée lors de l'envoi du premier paquet Hello mais sera ajoutée aux paquets Hello suivants suite à la réception du premier paquet Hello provenant du nœud voisin.
- *PGID* : le PGID du nœud.
- *Identifiant du port distant* : l'identifiant du port alloué par le nœud voisin permettant d'identifier le port physique et le VC décrit dans cette structure de données. Il ne sera pas diffusé lors de l'envoi du premier paquet Hello - le nœud n'ayant aucune connaissance de cette information à ce moment - mais sera diffusée dans les paquets Hello suivants suite à la réception du premier paquet Hello en provenance du nœud voisin.

Nous voyons donc que grâce au PGID et à l'identifiant de nœud véhiculés dans ce paquet Hello, deux nœuds se trouvant aux extrémités d'un lien considéré peuvent savoir si le nœud voisin fait ou non partie du même PG.

Lorsque deux nœuds voisins se rendent compte qu'ils n'appartiennent pas au même PG, les paquets Hello échangés par la suite contiennent des informations complémentaires dont le but est de déterminer un PG commun se trouvant au niveau hiérarchique le moins élevé possible. Chaque nœud va donc chercher dans sa base de données sur la topologie un nœud logique représentant son PG à un niveau plus élevé de la hiérarchie et se trouvant dans un PG commun avec le nœud logique représentant le PG de l'autre nœud au même niveau de la hiérarchie. Reprenons l'exemple de la Figure 4-4 et supposons qu'une machine Hello a été démarrée entre le nœud B.2.1 et le nœud B.1.1. Ces deux nœuds se rendent compte qu'ils n'appartiennent pas au même PG et échangent des informations dans le but de découvrir un PG commun. B.1.1 va alors dire à B.2.1 qu'il est représenté par le nœud B.1 dans le PG B et par le nœud B au sommet de la hiérarchie. De même, B.2.1 dit à B.1.1 qu'il est représenté par B.2 dans le PG B et par B dans le PG se trouvant en sommet de hiérarchie. Ces deux nœuds peuvent donc en conclure que leur PG commun de plus bas niveau est le PG B.

Ces informations, une fois diffusées par la procédure de synchronisation de base de données sur la topologie et par la procédure d'inondation, permettrons au PGL de B.2.1 et de B.1.1 de savoir qu'il doit exister un lien logique entre eux au niveau de hiérarchie supérieur.

II. Machine Hello aux autres niveaux de la hiérarchie

La logique générale de la machine Hello ayant été exposée à la section précédente, nous présenterons les différences principales qu'il existe lors de l'exécution d'une machine Hello à tout niveau de la hiérarchie (autre que le niveau physique).

Echange de données

Alors que l'échange des paquets Hello au niveau le plus bas de la hiérarchie se faisait au travers de VC réservés (VPI=0, VCI=18) appelés RCC, cet échange se fera aux travers de SVC établis par le module de signalisation du protocole PNNI entre deux nœuds logiques appartenant au même PG. Ces SVC sont également appelés RCC par analogie à ce qui se passe au niveau le plus bas de la hiérarchie.

Les paquets Hello échangés entre nœuds logiques ne contiennent pas les mêmes informations. La différence majeure entre la structure présentée au niveau le plus bas de la hiérarchie et celle qui s'applique à tous les autres niveaux est que l'on ne véhicule plus ni son PGID, ni le PGID du nœud voisin. Ces informations ne seraient d'aucune utilité vu que les RCC ont été établis uniquement entre des

nœuds logiques faisant partie du même PG. Il n'existe à ces niveaux de hiérarchie aucun lien sortant des PG et permettant l'exécution d'une machine Hello entre des nœuds frontière.

La structure du paquet Hello comprend également des informations à propos du SVC qui a été établi entre les deux nœuds logiques. Il s'agit principalement de timers permettant de détecter si après une certaine période d'inactivité sur ce SVC, il y a lieu de demander sa fermeture. Cette procédure n'existait pas au niveau le plus bas de la hiérarchie vu que les VC utilisés existaient aussi longtemps que les nœuds interconnectés étaient en état de marche.

Lorsque l'on se trouvait au niveau le plus bas de la hiérarchie et que deux nœuds constataient qu'ils n'appartenaient pas au même PG, ceux-ci rajoutaient des éléments d'information particuliers dans les paquets Hello. Ceci n'est plus le cas aux autres niveaux de la hiérarchie étant donné que, comme nous l'avons vu lorsque nous avons construit un réseau hiérarchique à la section 4.3.1a), il n'y a pas de connexions existantes entre les PG construits à tout autre niveau hiérarchique que le niveau physique.

ii - Synchronisation des bases de données concernant la topologie

La synchronisation des bases de données sur la topologie est le processus permettant à deux nœuds voisins d'avoir exactement la même base de données. Pour rappel, cette base de données sur la topologie est implémentée dans tous les nœuds du réseau et contient, au démarrage de chaque nœud, des informations connues directement par celui-ci (son adresse ATM, la description des liens qui sont connectés à ce nœud et l'ensemble des TE qui y sont connectés). Nous présenterons dans cette section le mécanisme global de la synchronisation.

La synchronisation des bases de données ne peut se faire qu'à partir du moment où, grâce à l'exécution des machines Hello, deux nœuds voisins ont pu conclure qu'ils appartiennent au même PG et se sont échangés toutes les informations nécessaires (se référer à la section précédente consacrée à la machine Hello).

A partir de ce moment, les nœuds entrent dans une période de négociation destinée à déterminer lequel des deux nœuds va être maître ou esclave de l'autre. Le but de cette relation maître/esclave est de déterminer lequel des deux nœuds va commencer la procédure de synchronisation.

Suite à la période négociation, chaque nœud va préparer un paquet de résumé de base de données (un *data summary packet*). Le paquet préparé doit contenir un identifiant de chaque PTSE contenu dans la base de données, identifiant décrivant le contenu de ce PTSE. Le paquet est ensuite envoyé par le nœud maître à travers le RCC vers le nœud voisin (l'esclave).

Le nœud esclave recevant un paquet de résumé de base de données en examine le contenu. Il compare chacun des identifiants des PTSE contenu dans ce paquet avec les PTSE qu'il maintient dans sa propre base de données. Tout identifiant de PTSE publié dans le paquet de résumé de base de données que le nœud ne possède pas ou qui est plus à jour que celui qu'il possède est introduit dans un paquet de requête de PTSE. Lorsque l'ensemble des identifiants de PTSE reçus dans le paquet de résumé de base de données ont été parcourus et examinés, le nœud esclave envoie le paquet de requête de PTSE ainsi que le paquet de résumé de base de données qu'il avait préalablement préparé.

Suite à la réception de ces deux paquets, le maître va envoyer les PTSE que le nœud esclave réclame et parcourir le paquet de résumé qu'il a reçu. Tout comme le nœud esclave, il va construire un paquet de requête de PTSE dans lequel figureront tous les identifiants de PTSE que le maître n'a pas ou ne possède pas en version à jour.

L'échange de paquets de résumé et de demande de PTSE se fait de manière identique jusqu'à ce que les deux nœuds aient une base de données identique. Ceci est détecté à partir du moment où, suite à l'envoi d'un paquet de résumé, le paquet reçu en retour contient exactement la même information.

Ajoutons pour conclure que nous pourrions définir la synchronisation des bases de données comme étant une procédure "photographique". En effet, supposant qu'une procédure de synchronisation a lieu entre

un nœud A et un nœud B et que, une fraction de seconde après le démarrage de cette procédure, une procédure similaire débute entre le nœud B et un nœud C, l'échange d'informations entre les nœuds B et A ne prendra en compte que les informations connues par B au moment précis où cette synchronisation a débuté. Toute information apprise par B à propos de C ne sera donc pas transmise à A durant la synchronisation, mais bien pendant la procédure d'inondation. Si cependant une synchronisation entre B et C avait déjà eu lieu au moment où la synchronisation entre B et A avait commencé, alors B donnerait également à A les informations reçues de C.

iii - Procédure d'inondation

Le rôle de la synchronisation de la base de données sur la topologie était d'assurer que deux nœuds voisins avaient la même base de données sur la topologie, constituée des mêmes PTSE.

Le rôle de la procédure d'inondation est d'assurer que *tous* les nœuds constituant un même PG ont la même base de données sur la topologie, donc constituée des mêmes PTSE.

La procédure d'inondation est démarrée par un nœud lorsque celui-ci a fini la période de synchronisation de base de données avec un de ses nœuds voisins et se poursuit jusqu'à la fermeture du lien (i.e. fermeture du RCC). Rappelons que, tout comme pour la période de synchronisation, l'inondation ne se passe qu'entre nœuds faisant partie du même PG.

Chaque nœud va préparer un paquet particulier, appelé *PNNI Topology State Packet* (PTSP), contenant les PTSE contenus dans la base de données sur la topologie. Ce paquet ne contient pas toujours l'ensemble complet des PTSE contenus dans cette base de données. Prenons par exemple le cas de deux nœuds logiques ne se trouvant pas au niveau le plus bas de la hiérarchie et appartenant au même PG. Ces deux nœuds ont donc ouvert entre eux un SVC devant servir de RCC. Nous avons dit que ce SVC était ouvert à l'aide du module de signalisation de PNNI, fortement basé sur le protocole UNI. Nous retrouverons dans ce module les mêmes concepts de négociation de contrat de trafic, QoS... Il incombera alors au nœud envoyant un PTSP de respecter le contrat de trafic qu'il a négocié et par conséquent, de veiller à ne pas envoyer de PTSP qui violerait ce contrat. Pour la suite nous considérerons que tous les PTSE du nœud considéré peuvent être contenus dans un seul PTSP.

Le PTSP préparé est envoyé à tous les nœuds voisins.

Lorsqu'un nœud reçoit un PTSP, il en examine le contenu. Tous les PTSE qu'il ne possède pas ou qui sont plus récents que ceux contenus dans sa propre base de données sont recopiés dans celle-ci. Suite à l'examen complet du PTSP reçu, le nœud prépare un paquet d'acquittement. Ce paquet contient l'identifiant de tous les PTSE qui ont été reçus et examinés et permet de signaler au nœud ayant envoyé le PTSP quels sont les PTSE qui ont été recopiés. Le paquet d'acquittement est enfin envoyé au nœud qui avait envoyé le PTSP.

Après avoir transmis ce paquet d'acquittement, le nœud ayant reçu le PTSP renvoie ce paquet à tous ses voisins, sauf celui par qui est arrivé le PTSP.

Après un certain laps de temps, tous les nœuds contenus dans le PG auront donc la même base de données. Ce mécanisme assure que tout changement intervenant à n'importe quel endroit de la topologie complète du réseau est répercuté à terme sur l'ensemble des nœuds du réseau mais à des niveaux d'agrégation différents : rappelons en effet que des informations précises sur la topologie d'un PG particulier seront diffusées au sein de ce PG mais qu'également des informations agrégées en provenance du nœud logique représentant ce PG à un niveaux de hiérarchie supérieur seront diffusées (donc par le PGL de ce PG).

iv - Mécanisme de résumé d'adresses

Si des informations concernant la topologie du réseau complet sont diffusées à travers celui-ci à des niveaux d'agrégation différents, il en est de même en ce qui concerne les données sur l'accessibilité (c'est-à-dire les adresses des TE connectés à l'ensemble des nœuds du réseau).

Basons-nous sur l'exemple de la Figure 4-7. Dans cette figure, les PG enfant des nœuds logiques Z.B et Z.C n'ont pas été représentés afin d'alléger le schéma.

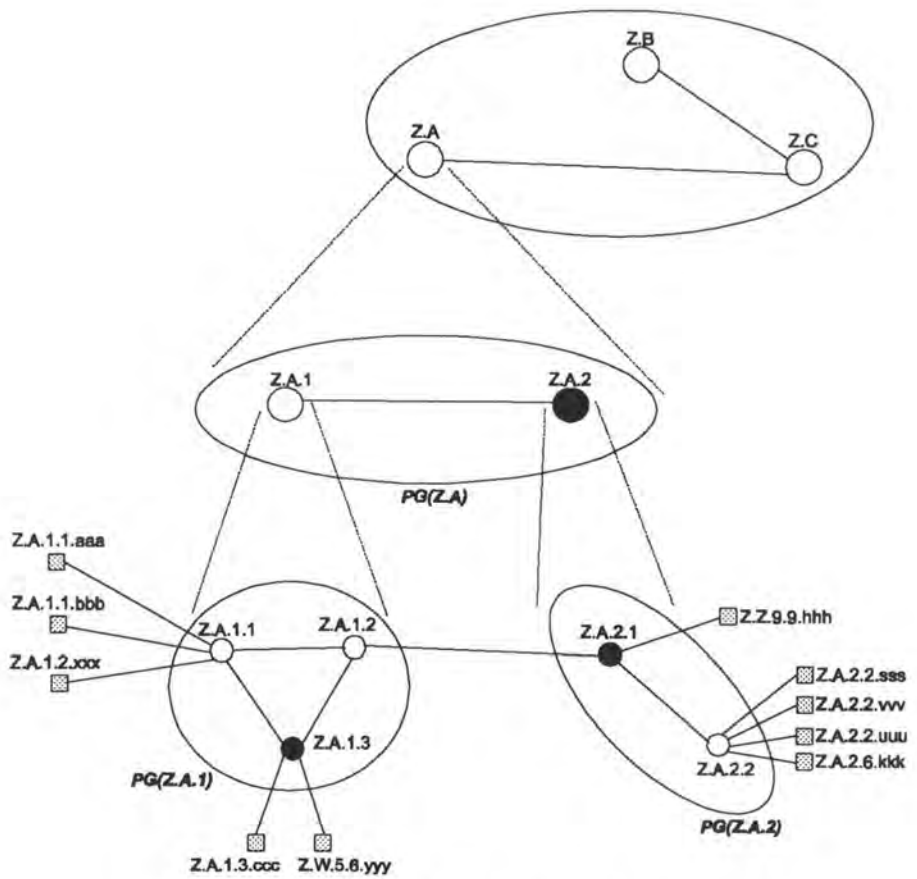


Figure 4-7 : exemple pour le résumé d'adresses

Prenons tout d'abord le nœud Z.A.1.1. 3 TE y sont connectés. Au lieu d'avertir l'ensemble de son PG qu'il a une connectivité sur ces trois machines, Z.A.1.1 va résumer ces adresses comme suit :

- les adresses Z.A.1.1.aaa et Z.A.1.1.bbb seront résumées en Z.A.1.1. Z.A.1.1 est l'adresse de résumé de ce nœud par défaut. Toute adresse de TE, telles que Z.A.1.1.aaa et Z.A.1.1.bbb, ayant le préfixe commun Z.A.1.1 avec le nœud sur lequel est connecté le TE ne sera donc pas publiée entièrement.
- l'adresse Z.A.1.2.xxx n'a pas de préfixe commun au moins égal à Z.A.1.1 correspondant à l'adresse de résumé par défaut du nœud Z.A.1.1. Cette adresse sera alors publiée dans son intégralité.

Le nœud Z.A.1.1 va donc publier les adresses résumées Z.A.1.1 et Z.A.1.2.xxx.

Prenons le nœud Z.A.1.3. Si le résumé d'adresse par défaut de ce nœud est Z.A.1.3, les résumés d'adresses publiés seront alors Z.A.1.3 et Z.W.5.6.yyy.

Ces résumés d'adresses sont publiés dans tout le PG Z.A.1. Le PGL Z.A.1.3, assumant également le rôle de nœud logique Z.A.1, récolte toutes ces données et, comme nous le savons, doit les diffuser sous forme agrégée dans le PG Z.A. Les adresses que Z.A.1 va devoir résumer sont :

- Z.A.1.1
- Z.A.1.2.xxx
- Z.A.1.3
- Z.W.5.6.yyy

L'adresse de résumé par défaut du nœud logique Z.A.1 est Z.A.1. Ce nœud logique va donc publier au sein du PG Z.A les adresses suivantes :

- Z.A.1
- Z.W.5.6.yyy

Suivant le même logique, le nœud Z.A.2 devra résumer les adresses suivantes :

- Z.A.2.2
- Z.A.2.6.kkk
- Z.Z.9.9.hhh

L'adresse de résumé par défaut du nœud logique Z.A.2 est Z.A.2. Il diffusera donc au sein du PG Z.A les adresses résumées suivantes :

- Z.A.2
- Z.Z.9.9.hhh

Nous savons que le nœud logique Z.A.1, qui est le PGL Z.A.1.3 du PG Z.A.1, doit diffuser au sein du PG Z.A.1 les informations agrégées qu'il a reçues dans le PG Z.A. Il va donc publier les résumés d'adresses qu'il aura reçu de Z.A.2. Grâce à cette information, tout nœud du PG Z.A.1 saura que pour établir une connexion avec un TE dont l'adresse commence par Z.A.2, il faudra passer par le nœud logique Z.A.2.

4.3.2.b) Sélection du chemin et Call Admission Control

Avant d'aborder la description du module de signalisation du protocole PNNI, terminons la section destinée au routage en décrivant deux mécanismes importants : la sélection du chemin à suivre pour une procédure de demande d'ouverture de connexion et le Call Admission Control dont le rôle est de vérifier à chaque nœud parcouru lors d'une demande d'ouverture de connexion que l'on peut bien offrir les ressources demandées par le TE initiateur de cette procédure.

i - Sélection du chemin

Une étape principale d'une procédure de connexion dans un réseau ATM est la sélection du chemin que la procédure d'appel devra suivre à partir du nœud jouant le rôle de point d'accès au réseau pour le TE appelant afin d'arriver jusqu'au nœud jouant le rôle de point d'accès au réseau du TE appelé. Le point

d'accès au réseau du TE appelant devra donc implémenter un algorithme de calcul de route à travers le réseau.

Il existe deux catégories principales de calcul des routes : le routage *hop-by-hop* et le routage à la source.

Le meilleur exemple de routage hop-by-hop est ce qui se passe dans le réseau Internet : lorsqu'un message doit être envoyé à travers le réseau, chaque nœud prend à son tour la décision du prochain nœud vers lequel le message doit être envoyé. Ce système est fortement consommateur de ressources système vu que le calcul doit être fait à chaque nœud traversé. De plus, un réseau ne transporte pas qu'un seul message à la fois, ce qui signifie que les ressources de temps de calcul doivent être partagées entre tous les messages arrivant à un nœud.

Un autre désavantage du routage hop-by-hop est l'apparition de boucles dans le chemin parcouru. Chaque nœud implémente en effet son propre mécanisme de calcul de routes et il n'est alors pas impossible que des nœuds déjà traversés soient à nouveau sélectionnés, comme l'illustre la Figure 4-8.

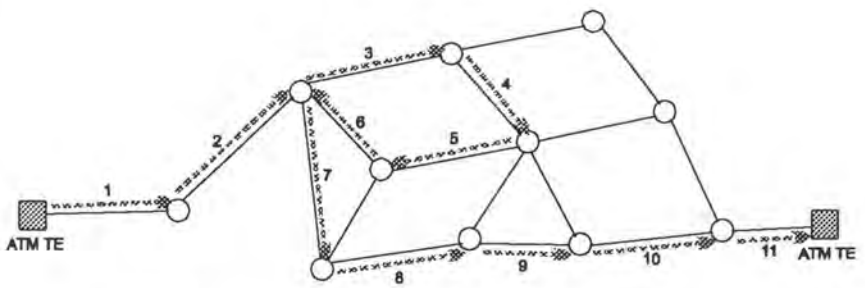


Figure 4-8 : exemple de bouclage avec un routage hop-by-hop

Le routage à la source permet d'éviter le problème de bouclage. Avec ce système, c'est le point d'accès au réseau du TE appelant qui va sélectionner la route complète que devra suivre le message de signalisation. Il suffit alors de programmer l'algorithme de routage de manière à ne pas permettre ce bouclage. Même si les nœuds constituant le réseau implémentent chacun un algorithme différent, cela ne posera aucun problème puisque la route aura déjà été calculée dans sa totalité.

Le désavantage de cette méthode est que chaque nœud doit avoir une vue complète de la topologie du réseau : toutes les données d'accessibilité ainsi que les caractéristiques précises des liens physiques doivent être connues.

Le protocole PNNI utilise un système de routage à la source. Mais comme nous l'avons vu, le but de la construction hiérarchique était précisément de pouvoir agréger les données de manière à ce qu'un nœud n'ait pas à garder une représentation précise du réseau. Le routage qui va être alors employé dans les nœuds d'un réseau ATM PNNI est appelé routage *partiel* à la source. Plutôt que de fournir une liste exacte de tous les nœuds physiques qui devront être traversés et de tous les liens physiques qui devront être utilisés, le rôle de l'algorithme de routage devra être capable de fabriquer une structure particulière appelé « pile de listes de transits désignés » ou DTL stack (DTL : *Designated Transit List*). Chaque liste se trouvant dans cette pile est appelée une DTL. Le pile de DTL sera ensuite transportée dans un message d'ouverture de connexion.

La structure d'une DTL peut être définie comme suit :

$$[NodeID_1(PortID_1), NodeID_2(PortID_2), ...];TransitPointer$$

Chaque DTL est donc une liste d'identifiants de nœud. A chacun de ces identifiants de nœud est attaché un identifiant de port[9].Cet identifiant indique le lien (physique ou logique) qui devra être utilisé pour transmettre le message de signalisation une fois que celui-ci sera arrivé au nœud considéré.

Le pointeur de transit ou *TransitPointer* figurant en queue de la DTL est utilisé afin d'indiquer le nœud sur lequel on se trouve, et ceci pour chacune des DTL se trouvant dans la pile.

Reprenant l'exemple du réseau exposé à la Figure 4-5 et supposant qu'une connexion doit être ouverte sur demande d'un TE entre le nœud A.1.1 et le nœud C.2.1, une pile de DTL qui pourrait être retournée par l'algorithme de routage exécuté sur le nœud A.1.1 ressemblerait à ceci :

```
[A.1.1(1), A.1.2(2)] ;1
[A.1(0), A.2(0)] ;1
[A(0), B(0), C(0)] ;1
```

Chacune des DTL constituant la pile donne une indication de routage à un niveau de la hiérarchie, la DTL se trouvant en sommet de pile faisant référence au niveau hiérarchique le plus bas, c'est-à-dire le niveau physique).

La pile de DTL que nous avons donnée en exemple doit être analysée de la façon suivante :

- la DTL en sommet de pile indique que l'on se trouve sur le nœud A.1.1 (le pointeur de transit vaut 1) et qu'il faut transmettre le message sur le port 1[10] afin de l'envoyer vers le nœud A.1.2 (i.e. le noeud suivant dans cette DTL).
- la deuxième DTL indique que l'on se trouve dans le nœud logique A.1 et que l'on doit aller vers le nœud logique A.2. Remarquons ici que les identifiants de port ont été mis à zéro. Zéro ne correspond à aucun port particulier. Ainsi le choix de port le mieux adapté afin d'aller vers A.2 sera laissé à la décision du nœud logique A.1. Les exemples de procédure de signalisation que l'on trouvera à la page 90 clarifieront ce mécanisme.
- la troisième DTL indique que l'on se trouve dans le nœud logique A et qu'il faut aller vers le nœud logique B.

Remarquons que les pointeurs de transit permettent, lorsque l'on regarde une pile de DTL, d'identifier un nœud à tous les niveaux de la hiérarchie.

Ceci n'explique toujours pas le terme de routage *partiel* à la source.

La construction hiérarchique que nous avons faite n'est qu'une forme de représentation du réseau. Son but était de pouvoir distribuer des informations agrégés sur certaines parties de ce réseau. Les SVC qui ont été établis entre les nœuds logiques à des niveaux de hiérarchie autres que le niveau physique n'avaient d'autre rôle que de véhiculer les informations résultant de l'exécution des machines Hello, des procédures de synchronisation des bases de données sur la topologie et des procédures d'inondation. Les messages de signalisation et par la suite les données utilisateur utiliseront non pas ces SVC particuliers mais les liens physiques du niveau hiérarchique le plus bas (il est clair que les SVC entre nœuds logiques utilisent les liens physiques du niveau hiérarchique le plus bas; cependant les messages de signalisation et les données utilisateur n'emploieront pas ces SVC).

Comme nous le voyons dans la pile de DTL donnée en exemple, seule la DTL se trouvant en sommet de pile décrit un chemin à emprunter au niveau physique. La deuxième DTL spécifie qu'après avoir traversé le nœud logique A.1 il faut aller vers le nœud logique A.2. Lorsque le message d'ouverture de signalisation arrivera dans le nœud logique A.2, la pile de DTL sera examinée. Grâce à un algorithme

---

[9] Un lien (physique ou logique) est associé à un port d'un nœud (physique ou logique). Chaque port est identifié par un identifiant de nœud qui est une valeur numérique.

[10] On suppose ici que le lien physique connectant les nœuds A.1.1 et A.1.2 est connecté à un port identifié par la valeur 1 dans le nœud A.1.1.

particulier (donné en annexe F), le nœud frontière du PG A.2 (représenté par le nœud logique A.2) ayant reçu le message saura qu'il faut maintenant se diriger vers le nœud logique B et son algorithme de routage rendra une nouvelle pile de DTL où la DTL se trouvant en sommet de pile décrira précisément le chemin à suivre dans ce PG A.2. La section consacrée aux scénarios de signalisation expose un exemple d'ouverture de connexion qui clarifiera l'utilisation et la génération des piles de DTL.

Nous pouvons donc en conclure que le routage partiel à la source est une forme de routage hybride entre un routage hop-by-hop et un routage à la source. Il n'aura cependant pas le désavantage du bouclage dans le système hop-by-hop étant donné que le pile de DTL est transportée dans le message d'ouverture de connexion et interdit au nœud calculant une route de repasser par un nœud déjà parcouru. Il n'aura pas non plus le désavantage du routage à la source vu que la topologie précise du réseau n'a pas dû être connue afin de tracer la route.

Ajoutons pour conclure que si le protocole PNNI spécifie le résultat que l'algorithme de routage doit retourner, il ne spécifie pas la manière dont la route doit être calculée. Ceci est laissé à la libre implémentation des constructeurs.

ii - Call Admission Control

Le Call Admission control (CAC) est une fonction devant être implémentée dans tout nœud faisant partie d'un réseau ATM PNNI. On distingue deux types de CAC : celui effectué à la source de la procédure d'appel, c'est-à-dire dans le point d'accès au réseau du TE appelant (cette fonction est alors appelée le *Generic Call Admission Control* ou *GCAC*) et le CAC effectué dans chaque nœud traversé lors de cette procédure.

Dans cette section, nous présenterons le rôle que le GCAC et le CAC doivent jouer. L'implémentation de ces deux fonctions ainsi que les paramètres précis devant être pris en compte et les calculs sur ces paramètres sortent du cadre de ce mémoire et ne seront pas exposés. Notons toutefois que le protocole PNNI ne définit pas comment le CAC doit être implémenté. Ceci est pour l'instant laissé au choix du constructeur.

I. Generic Call Admission Control

Lors du calcul initial de la route à suivre lors d'une procédure de connexion, c'est-à-dire dans le point d'accès au réseau du TE appelant, l'algorithme de routage doit pouvoir calculer une route traversant le réseau jusqu'au point d'accès au réseau du TE appelé (ou du moins jusqu'au nœud logique qui représente celui-ci à un niveau quelconque de la hiérarchie).

Sur base des informations agrégées reçues par le nœud "source" décrivant les liens physiques et logiques du réseau, le rôle du GCAC doit être de vérifier si un lien sélectionné par l'algorithme de routage est susceptible de pouvoir supporter les caractéristiques de trafic définies par l'utilisateur demandant l'ouverture de la connexion.

II. Call Admission Control

Lors d'une phase de demande d'ouverture de connexion, un nœud physique quelconque du réseau va recevoir le message de demande d'ouverture. Grâce à la pile de DTL contenue dans ce message, ce nœud va savoir quel est le lien qu'il doit utiliser afin de propager le message. Le rôle du CAC est alors de vérifier si le lien spécifié est capable de supporter les caractéristiques de trafic décrites dans ce message.

Si la demande de connexion peut être acceptée, les ressources demandées dans le message d'ouverture de connexion sont réservées pour cette connexion et le CAC doit répercuter cette réservation dans la base de données sur la topologie afin que les informations décrivant le lien emprunté soit toujours à jour. La demande de connexion sera ensuite propagée vers le nœud suivant.

Si les caractéristiques ne peuvent être supportées, un mécanisme particulier va être mis en cours. Ce mécanisme s'appelle le *crankback*. Au lieu que la demande d'ouverture de connexion soit rejetée, le crankback permet au nœud de regarder s'il existe une autre route permettant d'atteindre le point d'accès au réseau du TE appelé. Si une autre route ne peut être trouvée, un message particulier est envoyé au nœud par lequel est arrivé le message de demande d'ouverture. Ce nœud regarde alors à son tour s'il peut trouver une autre route et ainsi de suite.

Chaque nœud recevant le message particulier indiquant un mécanisme de crankback et calculant une nouvelle route fait appel à sa fonction CAC afin de vérifier que le lien susceptible d'être choisi peut supporter les caractéristiques de trafic définie dans le message d'ouverture de connexion qu'il avait gardé en copie.

L'appel sera rejeté si, en remontant l'ensemble du chemin déjà parcouru, aucun nœud n'a pu trouver une nouvelle route.

4.3.3 Module de signalisation

Le module de signalisation du protocole PNNI est, comme nous l'avons déjà dit, basé sur le protocole UNI 3.1 de l'ATM Forum. Le support de ce dernier protocole est même une règle principale du module de signalisation du protocole PNNI. Rappelons que PNNI est un protocole tournant exclusivement entre nœuds du réseau, mais à chaque nœud peut être attaché un ensemble de TE. Un nœud doit donc être capable d'effectuer des procédures de signalisation avec d'autres nœuds, mais aussi avec tout TE qui y est connecté. Le lecteur devra donc se rendre compte qu'il n'y a pas deux couches de signalisation dans un nœud, telles qu'une couche UNI 3.1 et une couche PNNI, mais une seule couche PNNI assurant les deux fonctionnalités. On pourrait alors dire que UNI 3.1 est « inclus » dans le module de signalisation de PNNI, mais que certaines fonctions ont été ajoutées afin de tirer profit de la construction hiérarchique du réseau. Ceci implique l'existence de nouveaux IE transportés dans les messages de signalisation destinés à supporter ces fonctionnalités (exemple : le transport de la pile de DTL et l'indication d'une procédure de crankback).

Au moment où est écrit ce mémoire, PNNI n'a pas encore été voté et approuvé définitivement par l'ensemble des membres de l'ATM Forum. PNNI est également régulièrement mis à jour suite aux discussions entre les membres de l'ATM Forum. Ainsi, suite à l'évolution de UNI 3.1 vers UNI 4, PNNI introduit au fur et à mesure de nouveaux concepts apparus dans UNI 4 (tels que le rajout d'une feuille à une connexion point-à-multipoint suite à l'initiative de la feuille, de nouveaux concepts de caractérisation de la QoS et de description du trafic, etc).

Cette section est basée sur la pré-version n°13 du protocole PNNI et ne fait référence qu'à des concepts de UNI 3.1 que nous avons vu au chapitre précédent.

Nous reprendrons la même structure d'exposé que celle utilisée au chapitre précédent. Plutôt que de tout redéfinir en détails, nous nous attacherons plus à marquer les similitudes et les différences avec UNI 3.1.

4.3.3.a) Contrôle d'appel

Dans le protocole UNI 3.1, nous avons défini la notion de contrôle d'appel. Du côté utilisateur de l'interface UNI, c'est-à-dire dans un TE, le contrôle d'appel était un programme particulier tel un pilote ou *driver* résidant dans la couche de signalisation et dont le rôle était d'utiliser les services offerts par le contrôle de protocole afin d'effectuer les demandes d'ouverture et de fermeture de connexion. Un utilisateur désirant ouvrir une connexion avec un utilisateur distant demandait l'ouverture d'une connexion par le biais du contrôle d'appel.

Le module de signalisation de PNNI, i.e. le contrôle de protocole, offre également ses services à un contrôle d'appel. La couche de signalisation est donc, tout comme pour le protocole UNI 3.1, constituée du contrôle de protocole et du contrôle d'appel. Le contrôle d'appel aura toujours pour but de gérer les demandes d'ouverture, fermeture et redémarrage d'une connexion, mais la notion d'utilisateur à qui reporter ces événements a disparu. Il s'agit donc d'un contrôle d'appel « automatisé » : suite à la réception d'une demande d'ouverture de connexion (par exemple) dans un noeud PNNI, le contrôle d'appel va demander directement au contrôle de protocole de propager la connexion jusqu'au nœud suivant (ou vers le TE appelé si ce nœud est le point d'accès de celui-ci), pour peu que les ressources demandées puissent être allouées.

4.3.3.b) Call States

i - Etats d'une procédure de connexion point-à-point

Tout comme pour le protocole UNI 3.1, on peut associer des états à une procédure de connexion point-à-point. Ces états reflètent les étapes parcourues lors d'une procédure de signalisation telles qu'une ouverture ou une fermeture de connexion.

Dans le chapitre précédent, l'exposé des différents états nous avait permis de déceler les phases principales de toute procédure de signalisation : demande d'ouverture et acceptation d'une connexion ainsi que fermeture de la connexion. Vu la compatibilité avec UNI que PNNI doit supporter, nous trouverons exactement les mêmes phases principales dans ce module de signalisation.

Rappelons-nous que, dans le protocole UNI, nous avions différentes classes d'état : celles associées à chacun des côtés de l'interface (l'utilisateur et le réseau) et celle associée au concept de référence globale. Le module de signalisation de PNNI ayant été dérivé de UNI 3.1, nous retrouverons également ces classes. Les notions des côtés utilisateur et côtés réseau de l'interface n'existent cependant plus étant donné que l'on ne considère plus que des communications entre nœuds du réseau. Cependant et afin de pouvoir différencier les deux côtés d'un lien connectant deux nœuds entre eux, on fera appel aux notions de **côté précédant** et de **côté succédant** d'un lien, comme l'illustre la Figure 4-9.

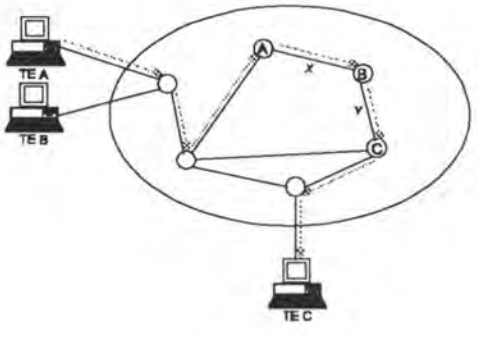


Figure 4-9 : illustration pour côté précédant et succédant d'un lien

Dans cette figure, une procédure de connexion entre le TE A et le TE C a lieu. Le chemin parcouru par cette demande de connexion est illustré par la flèche grise. Considérant le lien *X* et le sens dans lequel il est parcouru, le nœud A est le côté précédant de ce lien et le nœud B en est le côté succédant. Dans cette figure, le nœud B est donc le côté succédant du lien *X* et le côté précédant du lien *Y*.

Connaissant ces notions, nous pouvons à présent définir les états associés à une procédure de connexion. Ceux-ci se trouvent au Tableau 4-1.

Les états associés au concept de référence globale dans le protocole UNI se retrouvent également dans PNNI. Leur définition et la procédure à laquelle ils sont rattachés étant totalement identique à UNI 3.1, elles ne seront pas réexposées dans cette section.

Dénomination	*Label*	*Description*
Null	NN0	Pas d'appel en cours
Call Initiated	NN1	Etat existant au côté succédant d'un lien lorsque celui-ci a reçu une demande d'ouverture de connexion en provenance du côté précédant du lien mais n'y a pas encore répondu
Call Proceeding Sent	NN3	Etat existant au côté succédant d'un lien lorsque celui-ci a répondu à une demande d'ouverture de connexion
Call Present	NN6	Etat existant au côté précédant d'un lien lorsque celui-ci a envoyé une demande d'ouverture de connexion mais n'a pas encore reçu de réponse
Call Proceeding Received	NN9	Etat existant au côté précédant d'un lien lorsque celui-ci reçu une réponse suite à l'envoi d'une demande d'ouverture de connexion
Active	NN10	Etat existant aux deux côtés d'un lien lorsque la connexion est ouverte et disponible au trafic de données utilisateur
Release Request	NN11	Etat existant lorsqu'un nœud (qu'il fût côté succédant ou précédant d'un lien lors de l'ouverture de connexion) a envoyé une demande de fermeture de la connexion mais n'a pas encore reçu de réponse
Release Indication	NN12	Etat existant lorsqu'un nœud (qu'il fût côté succédant ou précédant d'un lien lors de l'ouverture de connexion) a reçu une demande de fermeture de la connexion mais n'y a pas encore répondu

Tableau 4-1 : liste des états associés à une procédure de signalisation

ii - Etats associés à une procédure point-à-multipoint

Les états associés à une procédure point-à-multipoint sont exactement semblables à ceux définis dans le chapitre précédent. Nous ne les redéfinirons donc pas dans ce chapitre.

4.3.3.c) Messages

On retrouve dans PNNI l'ensemble des messages que nous avions définis dans le chapitre précédent. Leurs définitions sont en tout point identiques : ils transportent globalement les mêmes informations, ont exactement les mêmes rôles et ont la même portée (i.e. globale vs locale) que dans UNI 3.1.

Nous retrouvons donc :

- pour la phase de demande d'ouverture : les messages SETUP et CALL PROCEEDING;
- pour la phase d'acceptation de connexion : les messages CONNECT et CONNECT ACKNOWLEDGE;
- pour la phase de fermeture ou de refus de connexion : les messages RELEASE et RELEASE COMPLETE;
- pour les demandes d'informations : les messages STATUS et STATUS ENQUIRY;
- pour la procédure de redémarrage : les messages RESTART et RESTART ACKNOWLEDGE;

- pour les procédures point-à-multipoint : les messages ADD PARTY, ADD PARTY ACKNOWLEDGE, ADD PARTY REJECT, DROP PARTY et DROP PART ACKNOWLEDGE.

Ces messages se présentent sous la même structure que les messages UNI. L'unique différence réside dans la valeur du discriminateur de protocole. Pour PNNI, celui-ci vaut, en binaire, 11110000.

Comme nous l'avions fait dans le chapitre consacré à UNI, la section 4.3.3.e exposera des scénarios de signalisation qui permettront de mettre en valeur l'utilisation de ces messages.

Nous avions mentionné au début de la section consacrée au module de signalisation de PNNI que certains IE allaient être ajoutés à des messages particuliers afin de transporter des informations propres au protocole PNNI (l'utilisation de la construction hiérarchique). Voyons quels sont ces éléments et dans quels messages ils doivent être transportés.

i - Pile de DTL

Lorsqu'un nœud reçoit une demande d'ouverture de connexion par un TE, nous savons qu'il doit calculer une route menant au point d'accès au réseau du TE appelé. Nous savons également que le fruit de ce calcul de route est une pile de DTL. Cette pile de DTL doit être transportée dans tout message d'ouverture de connexion afin d'être analysée dans chaque nœud traversé par la demande d'ouverture de connexion et doit permettre de savoir vers quel nœud le message reçu doit être envoyé. Les messages SETUP et ADD PARTY transporteront donc obligatoirement cette pile de DTL.

La pile de DTL est encodée dans un IE particulier appelé DTL IE. Celui-ci est exposé à l'annexe F.

ii - Indicateur de crankback

Nous avons brièvement exposé le principe du crankback à la section « Sélection du chemin et Call Admission Control ». Lorsqu'un nœud reçoit une demande d'ouverture de connexion (i.e. un message SETUP) mais qu'il ne peut supporter les caractéristiques de trafic définies dans ce message, nous avions dit que le nœud renvoyait alors vers le nœud qui lui avait transmis le message SETUP un message indiquant qu'une procédure de crankback devait être démarrée.

L'indicateur de crankback est codé dans un IE particulier appelé crankback IE (exposé à l'annexe F). Il est transporté dans les messages RELEASE et RELEASE COMPLETE.

Les messages RELEASE et RELEASE COMPLETE ont été choisis pour transporter cet IE étant donné le rôle joué par ceux-ci par définition, c'est-à-dire indiquer le refus d'une connexion.

Il est évident que lorsque ces messages sont utilisés afin de fermer une connexion sur demande d'un utilisateur l'IE crankback n'y est pas ajouté.

4.3.3.d) Primitives

Les primitives de services offertes par le module de signalisation du protocole PNNI sont identiques à celles que nous avons définies dans le chapitre précédent. Toutefois, deux nouveaux paramètres doivent être ajoutés : la pile de DTL et l'indicateur de crankback.

La pile de DTL doit être spécifiée lorsque l'on demande l'ouverture d'une connexion ainsi que lorsqu'il y a une indication de réception d'une demande d'ouverture de connexion. Nous la retrouvons donc dans les primitives SIG-OPEN.request, SIG-OPEN.indication, SIG-ADD_PARTY.request et SIG-ADD_PARTY.indication.

L'indicateur de crankback sera spécifié en paramètre dans toutes les primitives SIG-CLOSE pour peu qu'une procédure de crankback soit en cours. De même, on retrouvera ce paramètre dans les primitives

SIG-ADD_PARTY.resp et SIG-ADD_PARTY.confirmation si ces deux primitives sont utilisées afin de marquer le refus d'ajout de feuille à la connexion existante par manque de ressources disponibles au nœud ayant reçu cette demande d'ajout.

Les messages résultant de l'utilisation de ces primitives sont passés à la couche SAAL par l'intermédiaire de la primitive SSCF AAL-MESSAGE-FOR-TRANSMISSION.request(MU) où MU est le message passé par la couche de signalisation à la couche SAAL. De même, un message sera passé par la couche SAAL à la couche de signalisation par le biais d'une primitive SSCF AAL-RECEIVED-MESSAGE.indication(MU).

Tout comme nous l'avions fait au chapitre consacré à UNI, la section 4.3.3e exposera des scénarios de signalisation qui mettront en valeur l'utilisation des primitives.

4.3.3.e) Procédures

Nous exposerons dans cette section quatre scénarios de signalisation : une ouverture et une fermeture de connexion, une ouverture de connexion incluant une procédure de crankback et la création d'une connexion point-à-multipoint.

Avant de démarrer toute procédure de signalisation, nous retrouvons les mêmes procédures d'initialisation qui ont été exposées à la section 3.3.4 (scénarios de signalisation avec UNI 3.1), à savoir : l'initialisation de PNNI, la configuration du protocole, la connexion aux SAP et l'activation de la connexion.

Une fois ces procédures exécutées, l'entité de signalisation PNNI est prête à l'exécution de toute procédure de signalisation.

i - Ouverture d'une connexion

Reprenons le réseau que nous avons transformé en réseau hiérarchique au début de ce chapitre. Il est repris à la Figure 4-10 mais nous avons omis dans cette figure de représenter les niveaux de hiérarchie autres que le niveau physique. On consultera la Figure 4-5 pour avoir une illustration complète de tous les niveaux de hiérarchie.

Supposons que le TE d'adresse A.1.1.xxx connecté au nœud A.1.1 souhaite ouvrir une connexion avec le TE d'adresse C.1.2.ppp connecté au nœud d'adresse C.1.2.

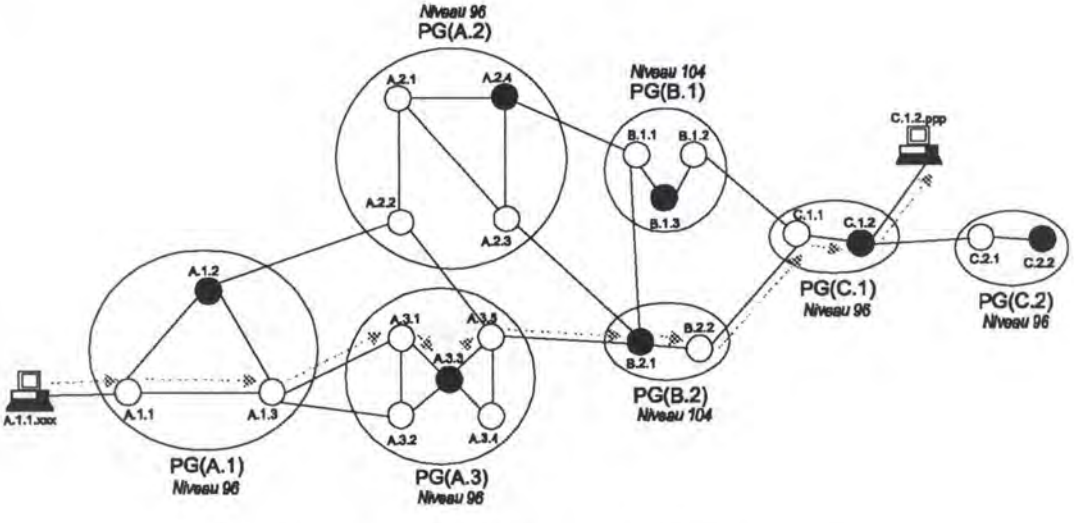


Figure 4-10 : chemin choisit par l'algorithme de routage

Les procédures de signalisation entre ces TE et leur point d'accès respectif au réseau sont totalement identiques à celles que nous avons exposées au chapitre 3, section 3.3.4.

I. Dans le nœud A.1.1

Supposons donc que le message de demande d'ouverture de connexion (message SETUP UNI) ait été reçu par le nœud A.1.1 auquel est connecté le TE A.1.1.xxx. Avant de demander un calcul de route sur base de l'adresse du TE appelé se trouvant dans ce message, le contrôle d'appel regarde si le message demande l'ouverture d'une connexion bidirectionnelle (il peut connaître cette information car rappelons-nous qu'il est possible de spécifier une QoS et un descripteur de trafic pour deux directions, l'une allant du TE appelant vers le TE appelé et réciproquement). Si c'est le cas, le contrôle d'appel fait appel au CAC du module de routage afin de vérifier que les ressources demandées peuvent être accordées sur le lien le reliant au TE A.1.1.xxx. Afin de simplifier cet exposé, nous supposerons par la suite qu'il n'existe aucun problème de disponibilité de ressources sur le réseau. Nous ne soulignerons donc plus les appels à la fonction CAC.

L'algorithme de routage du nœud A.1.1 retourne une pile de DTL destinée à router le message SETUP jusqu'au nœud C.1.2. Le chemin qui a été calculé doit respecter la QoS et les caractéristiques de trafic définies dans le message SETUP UNI reçu (ceci est vérifié par le GCAC). A partir des données agrégées sur l'accessibilité dont ce nœud dispose, il a pu se rendre compte que le TE appelé était joignable par le nœud logique C. La pile de DTL retournée sera par exemple :

[A.1.1(2), A.1.3(3)] ;1
[A.1(0),A.3(0)] ;1
[A(0), B(0), C(0)] ;1

Nous verrons dans la suite de cette section que le chemin qui sera suivi par le message de demande d'ouverture de connexion à travers le réseau sera celui illustré par les flèches grises de la Figure 4-10.

Dès que cette pile est rendue au contrôle d'appel, celui-ci va effectuer l'algorithme de traitement des DTL sur la pile. Le but de l'exécution de cet algorithme est de permettre au contrôle d'appel de savoir vers quel nœud il doit envoyer un message SETUP PNNI. La structure de cet algorithme est donnée à l'annexe F et servira de support à l'exposé.

La pile de DTL est traitée de la manière suivante : on regarde tout d'abord si le nœud actuellement pointé par le Transit Pointer dans la DTL de sommet de pile correspond bien au nœud A.1.1 à n'importe quel niveau de la hiérarchie. Ceci se fait en comparant l'identifiant du nœud A.1.1 à l'identifiant de nœud pointé par le Transit Pointer dans la DTL de sommet de pile[11].

Ici, A.1.1 correspond bien à A.1.1 à n'importe quel niveau de la hiérarchie et plus spécifiquement au niveau le plus bas de la hiérarchie. La DTL dont on dispose permet donc le routage à l'intérieur du PG de A.1.1. L'identifiant de port est sauvegardé dans une variable locale pour usage ultérieur.

L'élément de la DTL pointé n'étant pas le dernier élément de cette DTL, le Transit Pointer est incrémenté afin de pointer vers le nœud suivant. Etant donné que nous sommes arrivés en bout d'algorithme, on en ressort avec comme valeurs : l'identifiant de port sauvegardé en début d'algorithme, l'identifiant du nœud pointé par le Transit Pointer venant d'être incrémenté et la pile de DTL. La pile de DTL ressemble alors à :

---

[11] Pour comparer : si les niveaux des deux identifiants de nœud sont identiques, alors les deux nœuds sont normalement dans le même PG. Il faut donc que les adresses ATM des deux nœuds, contenues dans l'identifiant de ceux-ci, soient exactement égales. Si les niveaux ne sont pas égaux, alors pour qu'il y ait correspondance il faut que l'identifiant de nœud spécifié dans la DTL soit l'identifiant d'un nœud logique représentant le PG de A.1.1 à n'importe quel autre niveau de la hiérarchie. Il faut donc que l'on retrouve dans l'adresse ATM du nœud A.1.1 le préfixe composant l'identifiant du PG que représente le nœud pointé par le Transit Pointer (cet identifiant est encodé dans l'identifiant d'un nœud logique).

[A.1.1(2), A.1.3(3)] ;2
[A.1(0),A.3(0)] ;1
[A(0), B(0), C(0)] ;1

Le contrôle d'appel sait donc qu'il va devoir envoyer une demande d'ouverture de connexion au nœud A.1.3 à travers le port 2. Une primitive SIG-OPEN.request spécifiant les mêmes paramètres que ceux reçus dans le message SETUP UNI ainsi que la pile de DTL génère l'envoi d'un message SETUP vers le nœud A.1.3, comme illustré à la Figure 4-11.

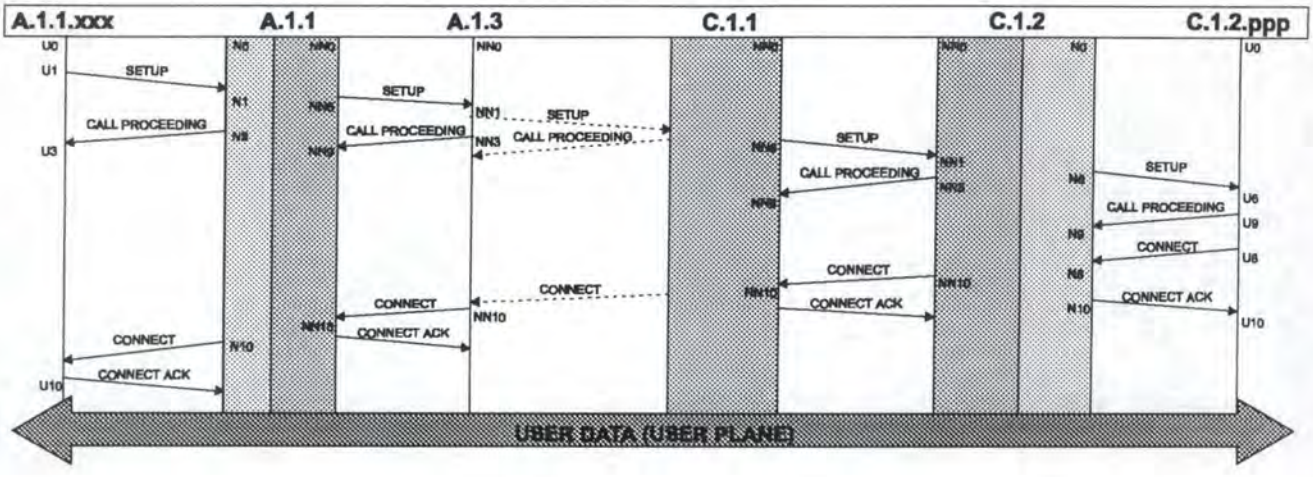


Figure 4-11 : flux de messages pour une procédure d'ouverture de connexion

II . Dans le nœud A.1.3

Un SIG-OPEN.indication prévient le contrôle d'appel du nœud A.1.3 de l'arrivée d'une demande d'ouverture de connexion. Les paramètres figurant dans ce message sont rendus au contrôle d'appel par cette primitive. Une primitive SIG-OPEN.response acquittera la réception de ce message par l'envoi d'un message CALL PROCEEDING. Le mécanisme d'allocation d'un couple (VPI,VCI) devant être spécifié dans ce message est alloué par l'entité de signalisation selon la même politique que UNI 3.1, à savoir : tout couple de valeurs (VPI,VCI) disponible[12].

Le contrôle d'appel récupère la pile de DTL et lui applique l'algorithme de l'annexe F.

Pour rappel, la pile de DTL reçue était :

[A.1.1(2), A.1.3(3)] ;2
[A.1(0),A.3(0)] ;1
[A(0), B(0), C(0)] ;1

Cette pile de DTL sera traitée de la manière suivante : regardant l'élément de la DTL de sommet de pile pointé par le Transit Pointer, on se rend compte que l'identifiant de nœud A.1.3 correspond bien au nœud A.1.3 sur lequel on se trouve et ceci au niveau le plus bas de la hiérarchie[13]. L'identifiant de port

---

[12] Il existe dans PNNI d'autres principes d'allocation de couple (VPI, VCI). On consultera à cet effet l'annexe H.

[13] Afin de comparer les deux identifiant de nœud (celui du nœud A.1.3 et celui fourni par la DTL) et tester s'ils font référence au même nœud physique , il suffit de voir que les deux identifiants spécifient le même niveau hiérarchique et la même adresse ATM.

attaché à l'identifiant de nœud pointé par le Transit Pointer (i.e. identifiant 3 pour le nœud A.1.3) est sauvé dans une variable locale current_port. Il s'agit ici du port (et donc par transition du lien physique) sur lequel sera envoyé le message SETUP.

L'élément pointé dans la DTL est le dernier élément de cette DTL. Etant donné que l'on est arrivé en bout de course de cette DTL et qu'elle n'est donc plus utile, elle est éliminée. La pile de DTL devient donc :

[A.1(0),A.3(0)] ;1
[A(0), B(0), C(0)] ;1

Ajoutons ici qu'il n'y aura que les nœuds frontière qui auront à supprimer une ou plusieurs DTL de la pile (après ce nœud, on sort en effet du domaine de routage du PG). Un tel nœud est appelé **exit border node** en regard du traitement des DTL.

Suivant l'algorithme de traitement des DTL, on examine ensuite la DTL suivante (devenue DTL de sommet de pile) et l'on regarde si le nœud pointé par le Transit Pointer de cette DTL correspond bien à A.1.3 à n'importe quel niveau de la hiérarchie. C'est le cas, A.1 étant le nœud logique représentant le PG de A.1.3 au niveau hiérarchique supérieur.

Chaque nœud logique est une représentation sous forme agrégé du PG qu'il représente. Le nœud A.1.3 doit pouvoir retrouver dans la description du nœud A.1 qu'il possède dans sa base de données sur la topologie (reçue par les mécanismes d'inondation) la liste des ports associés à ce nœud logique[14]. Si on ne peut trouver l'identifiant du port 3 (sauvegardé auparavant dans la variable current_port) dans cette liste alors, c'est que l'information sur la topologie a mal été distribuée et le processus est arrêté. Nous supposons par la suite que l'on peut trouver cet identifiant de port dans cette liste.

Examinant à nouveau la DTL, on voit que l'on ne se trouve pas sur le dernier élément de celle-ci. Le Transit Pointer est donc avancé vers l'élément suivant et l'on sort de l'algorithme en sachant que l'on doit envoyer un message SETUP partant sur le port 3 vers le nœud logique A.3.

La pile de DTL spécifiée dans le message SETUP envoyé suite à une primitive SIG-OPEN.request est donc :

[A.1(0),A.3(0)] ;**2**
[A(0), B(0), C(0)] ;1

Pour le traitement des DTL, le message SETUP est donc envoyé vers le nœud logique A.3. Cependant, pour le nœud A.1.3, ce message doit être envoyé sur le port d'identifiant 3. Ce port est connecté à un lien physique menant bien au noeud logique A.3, mais plus spécifiquement au noeud physique A.3.1 contenu dans celui-ci. Rappelons-nous en effet que le lien physique connectant les nœuds A.1.3 et A.3.1 avait été déclaré, au sein du PG du nœud A.1.3, comme étant un uplink vers l'upnode A.3.

III. Dans le nœud A.3.1

Un SIG-OPEN.indication prévient le contrôle d'appel du nœud A.3.1 de l'arrivée d'une demande d'ouverture de connexion. Les paramètres figurant dans ce message sont rendus au contrôle d'appel par cette primitive. Une primitive SIG-OPEN.response entraînant l'envoi d'un message CALL PROCEEDING acquittera la réception de ce message. Le mécanisme d'allocation du couple (VPI,VCI) sera le même que celui employé au nœud A.1.3. Par la suite, nous ne soulignerons plus cette allocation;

---

[14] Chaque PGL résume les informations qu'il a reçues de tous les nœuds de son PG. Ces informations concernaient entre autres la topologie. Lors de la construction de la représentation agrégée du PG sous forme d'un nœud logique, les identifiants des ports des nœuds frontière menant sur des liens sortant du PG sont repris dans une liste figurant dans la représentation agrégée du PG.

le lecteur devra toutefois se rendre compte que celle-ci est effectuée systématiquement dans tout nœud recevant un message SETUP.

La pile de DTL reçue dans les paramètres rendus par la primitive SIG-OPEN.indication est :

[A.1(0),A.3(0)] ;2
[A(0), B(0), C(0)] ;1

Le contrôle d'appel récupère cette pile et lui applique l'algorithme de traitement : l'identifiant de nœud A.3 pointé par le Transit Pointer dans la DTL de sommet de pile correspond bien au nœud A.3.1 à n'importe quel niveau de la hiérarchie (plus précisément A.3 est le nœud logique représentant le PG de A.3.1 au niveau de hiérarchie supérieur). L'identifiant de nœud A.3 n'est cependant pas égal bit à bit à l'identifiant du nœud A.3.1 et ne correspond donc pas à A.3.1 au plus bas niveau de la hiérarchie.

Etant donné que l'on sait que la DTL de sommet de pile ne spécifie pas une route au plus bas niveau de la hiérarchie, c'est-à-dire au sein du PG de A.3.1, cette route va devoir être calculée. Cependant et afin de calculer cette route, il est nécessaire de savoir vers quel nœud il faut se diriger. La DTL que l'on examine ne spécifiant plus d'éléments[15], l'algorithme va permettre de regarder la DTL de niveau (de pile) directement inférieur. Le but est de regarder au niveau hiérarchique supérieur vers quel nœud logique il faut se diriger. Dans notre cas, ce nœud logique est le nœud B.

Fournissant l'identifiant de ce nœud à l'algorithme de calcul de route, celui-ci fournira la DTL suivante : [A.3.1(3), A.3.3(2), A.3.5(4)] ;1.

Un nœud ayant à pratiquer une expansion de DTL (c'est-à-dire ajouter une nouvelle DTL en sommet de pile comme vient de le faire le nœud A.3.1) est appelé un **entry border node**.

Remarquons qu'en début d'algorithme de traitement des DTL, on sauvegarde l'identifiant de port pointé par le Transit Pointer de la DTL de sommet de pile. Avec la pile de DTL qui a été reçue, cet identifiant de port vaut 0. En fait, dès que l'algorithme de routage calcule une DTL qui ne se trouve pas au niveau le plus bas de la hiérarchie, il est conseillé de mettre systématiquement l'identifiant du port à 0. Si cette valeur avait été non nulle, cela signifiait que pour se diriger vers le nœud logique B on aurait dû obligatoirement sortir du nœud A.3.1 par ce port. Or, souvenons-nous que la pile de DTL originale a été calculée par le nœud A.1.1 qui n'avait qu'une connaissance agrégée de la topologie du PG A.3. En laissant le port à 0 on permet alors à l'algorithme de routage du nœud A.3.1 de calculer une route au travers du PG de A.3.1 en ne forçant pas un port de sortie dès l'origine. Etant donné que le nœud A.3.1 a une connaissance plus précise de la topologie de son PG, celui-ci peut très bien choisir de sortir par un port menant sur un lien physique plus avantageux que celui qui aurait été choisi par le nœud A.1.1.

La DTL rendue par l'algorithme de routage du nœud A.3.1 signifie donc que pour arriver dans le nœud logique B, il faut traverser les nœuds A.3.1, A.3.3 et A.3.5.

Cette DTL est ajoutée en sommet de pile.

A.3.1 n'étant pas le dernier élément de cette nouvelle DTL, le Transit Pointer est avancé pour pointer sur la prochaine destination (A.3.3). La nouvelle pile de DTL, l'identifiant du nœud vers lequel on doit se diriger (A.3.3) et l'identifiant du port de sortie (3) sont rendus en sortie d'algorithme.

La pile de DTL est donc maintenant :

[A.3.1(3), A.3.3(2), A.3.5(4)] ;**2**
[A.1(0),A.3(0)] ;2
[A(0), B(0), C(0)] ;1

Elle est donnée en argument à la primitive SIG-OPEN.request avec les autres paramètres reçus dans le message SETUP et laissés inchangés.

---

[15] S'il y avait encore un élément dans la DTL après A.3, c'est vers cet élément qu'il faudrait se diriger.

L'utilisation de cette primitive génère l'envoi d'un message SETUP vers le nœud A.3.3.

L'échange de messages SETUP et CALL PROCEEDING est le même entre tous les nœuds du réseau. Nous nous concentrerons à partir de maintenant sur la propagation du message SETUP en fonction du traitement de la pile de DTL et ne soulignerons plus l'utilisation des primitives et l'envoi des messages de confirmation CALL PROCEEDING. Il est clair cependant que ceci est toujours effectué.

Jusqu'au nœud A.3.5, l'unique traitement sur cette pile sera d'avancer le Transit Pointer de la DTL de sommet de pile.

IV. Dans le nœud A.3.5

La pile de DTL reçue par le nœud A.3.5 dans le message de SETUP est :

[A.3.1(3), A.3.3(2), A.3.5(4)] ;**3**
[A.1(0),A.3(0)] ;2
[A(0), B(0), C(0)] ;1

Suite aux vérifications classiques (le nœud pointé correspond bien au nœud A.3.5 à n'importe quel niveau de la hiérarchie et au plus bas niveau de la hiérarchie), le traitement suivant est effectué :

- suppression de la DTL de sommet de pile (on est au bout);
- suppression de la nouvelle DTL de sommet de pile [A.1(0),A.3(0)] ;2 (on est au bout);
- avancement du Transit Pointer sur le deuxième élément de la dernière DTL.

Le message SETUP quittant le nœud A.3.5 vers le nœud logique B (plus précisément vers le nœud physique B.2.1 : l'identifiant de port 4 spécifié dans la DTL de sommet de pile initialement reçue mène vers ce nœud) est donc :

[A(0), B(0), C(0)] ;**2**

V. Dans le nœud B.2.1

Le nœud B.2.1 ayant reçu un message SETUP contenant cette dernière pile de DTL la traite de la manière suivante :

- B correspond bien à B.2.1 à n'importe quel niveau de la hiérarchie mais pas au niveau physique;
- le prochain nœud vers lequel il faut se diriger est le nœud logique C;
- L'algorithme de routage du nœud B.2.1 va rendre deux DTL : l'une pour le niveau le plus bas de la hiérarchie, c'est-à-dire le niveau du nœud B.2.1 et l'autre pour le niveau intermédiaire entre le niveau de B.2.1 et le niveau du nœud logique B (on ne peut pas avoir des niveaux manquants entre la DTL de sommet de pile et la DTL de fin de pile). Les DTL rendues par l'algorithme de routage du nœud B.2.1 sont alors :

  [B.2.1(2), B2.2(1)] ;1
  [B2(0)] ;1

- le Transit Pointer de la DTL de sommet de pile est avancé vers l'élément B.2.2 de la DTL de sommet de pile.

B.2.1 va donc envoyer un message SETUP vers B.2.2 contenant la pile de DTL suivante :

[B.2.1(2), B2.2(1)] ;2
[B2(0)] ;1
[A(0), B(0), C(0)] ;2

VI. Dans le nœud B.2.2

B.2.2 est un exit border node au yeux de l'algorithme de traitement des DTL. Etant donné que l'on est en bout de course de la DTL de sommet de pile, celle-ci est supprimée.

La deuxième DTL est également supprimée étant donné que l'on est en bout de course de cette DTL et que l'on sort du domaine de routage de B.2.

Le Transit Pointer de la dernière DTL est avancé sur le dernier élément.

B.2.2 va donc envoyer un message SETUP vers le nœud logique C (plus précisément vers le nœud C.1.1) via le port identifié par 1 (tiré de la DTL de sommet de pile reçue). Le message SETUP envoyé contiendra la pile de DTL suivante :

[A(0), B(0), C(0)] ;3

VII. Dans le nœud C.1.1

Le nœud C.1.1 reçoit donc un message SETUP contenant une pile de DTL dans laquelle n'existe plus qu'une seule DTL.

Les vérifications d'entrée d'algorithme de traitement des DTL permettent d'établir que l'identifiant de nœud C pointé par le Transit Pointer de cette DTL correspond bien au nœud C.1.1 à n'importe quel niveau de la hiérarchie mais pas au niveau physique.

Etant donné que l'on se trouve sur le dernier nœud pointé par le Transit Pointer de la DTL et qu'il n'y a pas d'autres DTL dans la pile, on peut en conclure que l'on se trouve dans un nœud logique contenant le nœud physique auquel est connecté le TE appelé. On fait alors appel à l'algorithme de routage en lui spécifiant l'adresse ATM exacte du TE appelé.

Les DTL suivantes seront rendues par l'algorithme de routage :

[C.1.1(3), C.1.2(0)] ;1
[C.1(0)] ;1

Ces DTL sont ajoutées en sommet de la pile actuelle (qui ne contenait qu'une seule DTL) et le Transit Pointer de la nouvelle DTL de sommet de pile est avancé.

En sortie d'algorithme de traitement des DTL, le contrôle d'appel sait qu'il faut demander une ouverture de connexion avec le nœud C.1.2 et que le lien utilisé pour envoyer le message SETUP résultant de cette demande est connecté au port d'identifiant égal à 3. Ce message SETUP contiendra la pile de DTL suivante :

[C.1.1(3), C.1.2(0)] ;2
[C.1(0)] ;1
[A(0), B(0), C(0)] ;3

VIII. Dans le nœud C.1.2

C.1.2 analyse la DTL qu'il a reçue dans le message SETUP et y applique l'algorithme de traitement :

- la DTL de sommet de pile est supprimée (on est à la fin de cette DTL);
- la deuxième DTL est également supprimée (idem);
- la dernière DTL est supprimée (idem).

Etant donné qu'il ne reste plus aucune DTL dans la pile, c'est que l'on se trouve dans le nœud auquel est connecté le TE appelé. Le nœud C.1.2 utilise alors les procédures de signalisation UNI pour transmettre la demande d'ouverture de connexion à ce TE (illustré à la Figure 4-11).

IX . Pour conclure

La notion de routage partiel à la source est maintenant claire : une route plus ou moins grossière est générée par le premier nœud impliqué dans la procédure de connexion. Ensuite, à chaque fois que l'on entre dans un nouveau nœud logique, un nouvelle route à travers ce nœud est calculée et vient préciser la route initiale. On comprend mieux maintenant cette combinaison de routage à la source et de routage hop-by-hop constituant la méthode de routage partiel à la source.

ii - Fermeture de connexion

La procédure de fermeture de connexion entre nœuds d'un réseau est en tout point similaire à celle exécutée avec le protocole UNI 3.1 entre un TE et son point d'accès au réseau, nous ne la réexpliquerons donc pas en détails.

La Figure 4-12 illustre la fermeture de la connexion que nous venions d'établir à la section précédente. Cette fermeture se fait, dans notre exemple, sur demande du TE A.1.1.xxx selon les procédures que nous avions exposées à la section 3.3.4b.

L'entité de signalisation du nœud A.1.1 recevant ce message génère un SIG-CLOSE.indication auprès du contrôle d'appel de ce nœud. La demande de fermeture de connexion est alors propagée vers le nœud suivant sous forme d'un message RELEASE. La réception de ce message est signalée par un SIG-CLOSE.request et la connexion est fermée avec ce nœud. Il s'agit donc d'une déconnexion progressive de la connexion : tous les SVC établis entre chaque paire de nœuds prenant part à la connexion sont fermés les uns après les autres suivant la progression du message de déconnexion à travers le réseau.

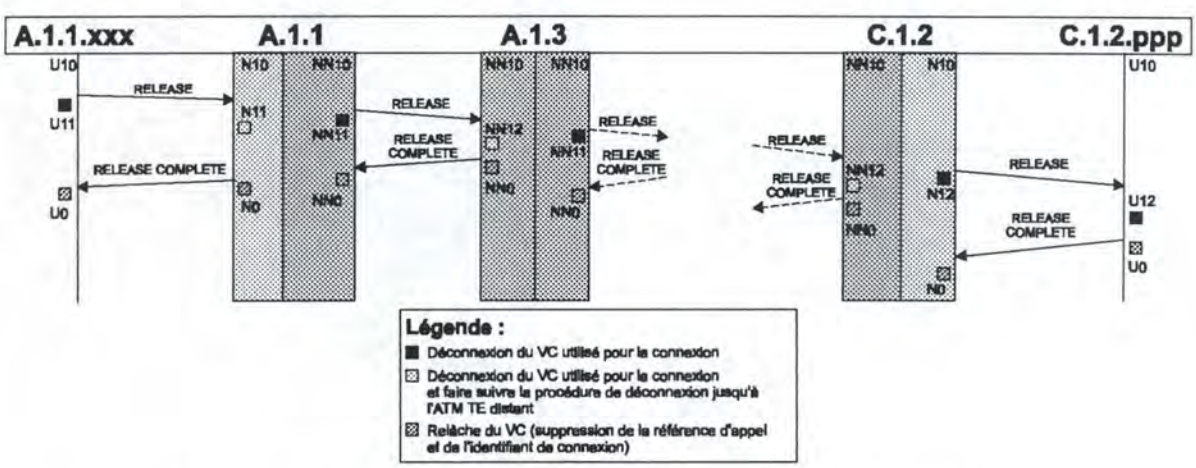


Figure 4-12 : flux de messages pour une procédure de fermeture de connexion

iii - Deux cas de crankback

Nous avons présenté dans ce chapitre une procédure de connexion où aucun problème ne s'était passé. Nous avions supposé que les ressources réseaux étaient toujours disponibles.

La section consacrée au CAC a introduit le concept de crankback selon lequel une demande de connexion ne pouvant être acceptée par un nœud pour des questions de disponibilité de ressources pouvait être renvoyée en arrière afin de trouver un autre chemin menant au TE appelé.

On distingue en fait 3 blocages générant une procédure de crankback :

1. le blocage dans un nœud : on dit qu'il y a blocage dans un nœud dans 2 cas :
   - il ressort du traitement effectué sur la pile de DTL reçue dans le message SETUP que le nœud sur lequel on se trouve est le nœud sur lequel est censé être connecté le TE appelé. Cependant ce TE n'y est pas connecté;
   - il ressort du traitement effectué sur la pile de DTL que le message SETUP doit être propagé au nœud suivant. L'identifiant du nœud suivant et le port menant à ce nœud ont été rendus par ce traitement. Cependant le nœud sur lequel on se trouve n'a aucun connaissance d'une connectivité avec le nœud identifié par le traitement de la pile de DTL;
2. le blocage au côté précédant d'un lien : on dit qu'il y a blocage au côté précédant d'un lien si, avant d'envoyer le message SETUP au nœud suivant, il ressort du CAC du nœud que le lien ne peut supporter la QoS ou les caractéristiques de trafic définies dans ce message, ou encore que ce lien est hors d'usage (problème technique).
3. le blocage au côté succédant d'un lien : on dit qu'il y a blocage au côté succédant d'un lien si, après avoir reçu une demande de connexion (i.e. un message SETUP) spécifiant l'ouverture d'une connexion bidirectionnelle, il ressort du CAC du nœud que le lien menant au nœud précédent ne peut supporter la QoS ou les caractéristiques de trafic définies pour la direction "nœud appelé vers nœud appelant".

Ces définitions étant posées, étudions un premier cas de procédure de crankback.

1. Premier cas

Considérons l'exemple de la Figure 4-13. Il est basé sur l'exemple d'ouverture de connexion qui avait été exposé à la page 90.

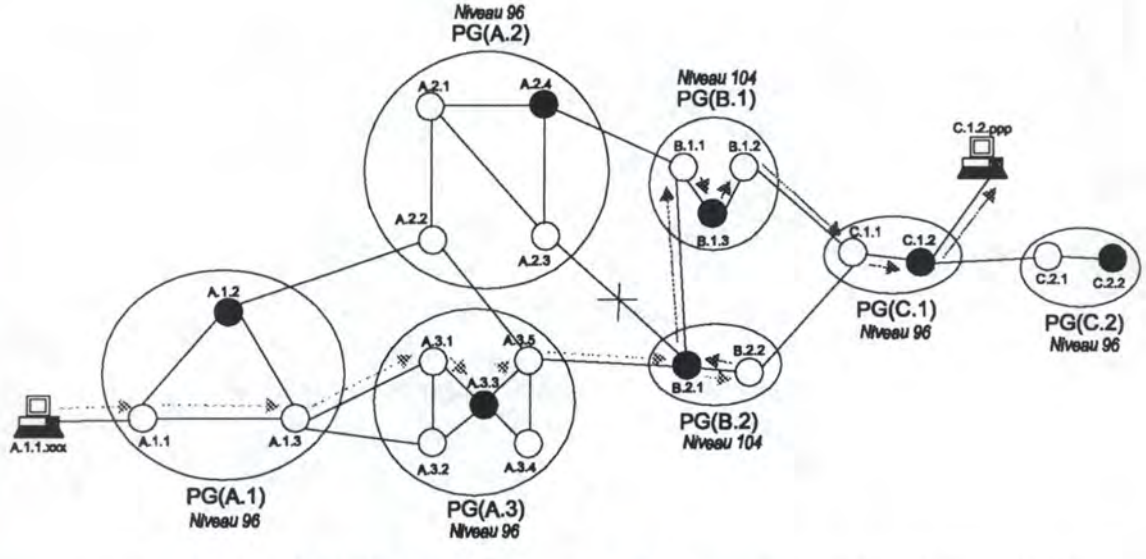


Figure 4-13 : chemin choisi pour le message SETUP avec une procédure de crankback

Supposons que le message SETUP a été propagé jusqu'au nœud B.2.2 et que ce message demande l'ouverture d'une connexion bidirectionnelle. Cependant le CAC du nœud B.2.2 indique que le lien B.2.2-B.2.1 ne peut supporter les caractéristiques de trafic définies dans le message SETUP pour la direction nœud appelé vers nœud appelant.

Dans ce cas, le nœud B.2.2 va envoyer un message RELEASE au nœud B.2.1 transportant une indication de procédure crankback et précisant que le blocage a eu lieu au côté succédant du lien.

Suite à la réception de ce message, le contrôle d'appel de B.2.1 va vérifier si des DTL ont été ajoutées au message SETUP lorsque celui-ci est arrivé dans ce nœud. Tout nœud ayant ajouté des DTL doit garder une copie de la pile de DTL originale et une copie des DTL qu'il a rajoutées et ceci jusqu'à ce qu'il soit notifié que l'appel est accepté (donc jusqu'à la réception d'un message CONNECT). Il est donc facile de savoir si des DTL ont été rajoutées ou non. Le but de ce test est de voir si le nœud était un entry border node pour cette procédure de connexion. Dans ce cas, cela signifie qu'il a dû calculer une route à travers son PG. Sans doute ce nœud peut-il trouver une autre route permettant de ne pas passer par B.2.2 ?

Dans notre exemple, deux routes pourraient exister.

L'une serait de partir de B.2.1 vers le PG B.1 et enfin arriver dans le PG C.1. Rappelons-nous que la DTL reçue par B.2.1 était :

[A(0), B(0), C(0)] ;**2**

La nouvelle pile de DTL pour passer par le nouveau chemin serait alors :

[B.2.1(1)] ;1
[B2(0),B1(0)] ;1
[A(0),B(0),C(0)] ;2

Cette route pourrait être sélectionnée car elle ne fait pas apparaître de boucle dans la route choisie et respecte la pile de DTL qui a été originalement reçue (ceci signifie que les DTL de la pile de DTL reçue ne doivent pas être modifiées - on ne fait que rajouter des DTL). Ce respect de la pile de DTL reçue est une condition devant toujours être respectée lors d'une procédure de crankback.

L'autre route serait de partir de B.2.1 vers le PG A.2 pour passer ensuite par le PG B.1 et enfin le PG C.1. Cependant cette route ne peut pas être prise car la nouvelle pile de DTL serait alors :

[B.2.1(1)] ;1
[B2(0)] ;1
[A(0),B(0),A(0),B(0),C(0)] ;2

Or les boucles dans le routage ne sont pas permises et l'on voit bien que la DTL à la base de la pile obligerait de repasser par les nœuds logiques A et B (de plus cette DTL aurait du être modifiée, entraînant un non-respect de la pile de DTL reçue). La première route est alors choisie.

La procédure de connexion qui s'était arrêtée en B.2.1 peut donc continuer. Le contrôle d'appel de B.2.1 utilise la primitive SIG-OPEN.request pour envoyer un message SETUP contenant les paramètres délivrés par la primitive SIG-OPEN.indication lors de la réception du message SETUP en provenance de A.3.5 et incluant le nouvelle pile de DTL qui, après traitement, sera donc :

[B2(0),B1(0)] ;2
[A(0),B(0),C(0)] ;2

Pour la suite de la connexion, on supposera qu'il n'y a plus de problèmes de ressources. La procédure de connexion se déroulera donc de manière similaire à l'exemple donné à la page 90.

II. Deuxième cas

Dans le premier cas, nous avions supposé que le lien B.2.1-B.1.1 pouvait supporter la nouvelle connexion. Considérons maintenant que ce lien est hors d'usage. B.2.1 n'a donc plus aucune possibilité de rerouter la demande de connexion. Il continue la procédure de crankback en envoyant au nœud précédent (i.e. A.3.5) un message RELEASE incluant un indicateur de crankback mentionnant un blocage au sein du PG B.2 (ceci afin que le nœud logique représentant ce PG ne soit plus sélectionné lors de routage ultérieur pour le même appel).

Le nœud A.3.5, n'ayant ajouté aucune DTL (il était exit border node), n'a pas le droit de prendre de nouvelles décisions de routage et le message RELEASE est propagé au nœud A.3.3. Remarquons que le message RELEASE a le même effet que lors d'une procédure de déconnexion classique, à savoir libérer les couples (VPI, VCI) et la référence d'appel qui avait été alloués pour le lien menant de A.3.5 à B.2.1 (par exemple).

A.3.3 n'étant pas entry border node, il ne peut participer au reroutage et le message RELEASE est envoyé au nœud A.3.1.

A.3.1 est un entry border node pour cette procédure de connexion, il peut participer au reroutage. La seule route qui pourrait être trouvée serait d'aller jusqu'au nœud A.3.5, de passer dans le PG A.2, puis par le PG B.1 pour ensuite arriver dans le PG C.1. La pile de DTL résultante serait alors :

[A.3.1(3), A.3.3(2), A.3.5(4)] ;1
[A.1(0),A.3(0),A.2(0)] ;2
[A(0), B(0), C(0)] ;1

Or, la pile de DTL reçue par ce nœud était :

[A.1(0),A.3(0)] ;2
[A(0), B(0), C(0)] ;1

On voit donc que si le nouveau chemin était choisi, la pile de DTL reçue ne serait pas respectée (il faudrait modifier la deuxième DTL). Par conséquent, un message RELEASE indiquant un crankback est envoyé au nœud A.1.3.

A.1.3 ne pouvant participer au reroutage (il est exit border node), le RELEASE est propagé jusqu'au nœud A.1.1.

A.1.1 était le nœud d'origine de la procédure d'appel. Il n'a aucune restriction pour chercher un nouveau chemin étant donné qu'il n'a pas de pile de DTL à respecter[16].

Le nouveau chemin pourrait alors consister à partir vers le nœud A.1.2, puis vers le nœud logique A.2, puis B, puis C. La nouvelle pile de DTL serait alors :

[A.1.1(3),A.1.2(2)] ;1
[A.1(0),A.2(0)] ;1
[A(0), B(0), C(0)] ;1

A.1.1 envoie alors un message SETUP vers le nœud A.1.2 avec la nouvelle pile de DTL et la procédure de connexion peut continuer comme nous l'avons déjà expliquée.

---

[16] Notons qu'il existe toutefois une restriction, celle de ne pas passer par le nœud logique B.2. Cette information est tirée du crankback IE spécifié dans le message RELEASE reçu.

iv - Ajout d'une feuille à une connexion point-à-multipoint

Les échanges de messages entre nœuds lors de l'ajout d'une feuille sont en tout point similaires à ceux prenant place entre un TE et son point d'accès au réseau dans les procédures UNI. Nous ne les réexpliquerons pas ici.

Nous nous concentrerons dans cette section sur la notion de dernier nœud commun (LCN : *last common node*), de transformation des messages ADD PARTY en message SETUP, d'allocation d'identifiant de connexion et de référence d'appel.

Considérons la Figure 4-14.

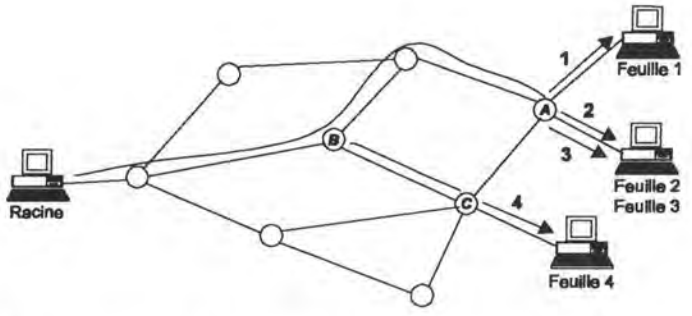


Figure 4-14 : ajout de 4 feuilles pour une connexion point-à-multipoint

Dans cette figure, on souhaite établir une connexion point-à-multipoint entre un TE racine et 4 feuilles.

I. Première feuille

La connexion avec la première feuille se fait par la procédure exposée à la section similaire dans chapitre 3. La demande d'ouverture de connexion est donc transportée dans un message SETUP et passe à travers le réseau selon les procédures exposées précédemment dans ce chapitre (DTL, crankback si nécessaire, ...).

II. Deuxième feuille

L'ajout de la deuxième feuille se fait par l'envoi d'un message ADD PARTY à partir du TE racine. Ce message n'est pas transformé en message SETUP tant que le chemin parcouru pour joindre la deuxième feuille est identique à celui parcouru pour l'ajout de la première feuille. Dans le cas de la Figure 4-14, la feuille 2 étant connectée au même nœud A que la feuille 1, le message d'ajout d'une feuille restera un message ADD PARTY jusqu'à ce nœud. Il aura la même référence d'appel et le même identifiant de connexion que le message SETUP initial.

Par contre, une fois arrivé au nœud A, aucune connexion n'existe jusqu'à la feuille 2. Pour cette raison, le message ADD PARTY est transformé en message SETUP et envoyé à la feuille 2. Celle-ci sait qu'il s'agit d'une connexion point-à-multipoint dû à la présence d'une référence de point terminal associé à cette feuille (référence définie par la racine et véhiculée dans le message ADD PARTY) et l'indication d'une configuration point-à-multipoint dans l'IE Broadband Bearer Capability transporté dans ce message. L'identifiant de connexion sera, pour le lien entre le nœud A et la feuille 2, différent que celui utilisé jusqu'alors (on diverge du chemin initial) et il sera alloué par le nœud A.

Le nœud A est ici le LCN pour les feuilles 1 et 2. Toutes les données en provenance de la racine seront dupliquées en ce nœud, une copie étant envoyée vers la feuille 1 et l'autre à la feuille 2.

III. Troisième feuille

Dans l'exemple que nous donnons ici, la troisième feuille se trouve dans le même TE que la deuxième feuille. Il se peut que l'utilisateur de ce TE désire, pour une raison qui lui est propre, jouer deux fois le rôle de "client".

Le chemin suivi par le message de demande d'ajout de feuille ADD PARTY afin de joindre la troisième feuille emprunte le même chemin que pour la première et la deuxième feuille.

Une fois que la demande d'ajout atteint le nœud A, celui-ci se rend compte qu'il a déjà une connexion similaire ouverte avec la feuille 2 dans le même TE. Dans ce cas, le message ADD PARTY n'est pas converti en message SETUP et envoyé tel quel à ce TE.

Dans le cas de la Figure 4-14, le LCN pour les feuilles 1 et 2 reste le nœud A, mais le LCN pour les feuilles 2 et 3 est le TE. C'est donc l'entité de signalisation de ce TE qui dupliquera les données reçues pour les fournir aux deux applications clientes.

Le message ADD PARTY reçu par l'entité de signalisation du TE supportant les feuilles 2 et 3 n'a donc d'autre but que prévenir cette entité que toute donnée reçue sur le VC établi lors de la création de la deuxième feuille doit être également délivrée à une autre application, identifiée par l'adresse ATM contenue dans le message ADD PARTY.

IV. Quatrième feuille

Pour l'ajout de la quatrième feuille, le chemin parcouru reste identique jusqu'au nœud B. Le message de demande d'ajout d'une feuille ADD PARTY envoyé par la racine restera donc sous cette forme jusqu'à ce nœud.

Cependant, après le nœud B, le chemin restant à parcourir est différent de celui pris jusqu'à présent. Etant donné que l'on ajoute une nouvelle « branche » à la connexion point-à-multipoint, le message est transformé en un message SETUP et les techniques de signalisation PNNI sont utilisées pour mener le message jusqu'au nœud C. Le descripteur de trafic, la QoS et les autres paramètres du messages SETUP sont identiques à ceux reçus par le nœud B pour l'ajout de la première feuille.

Une fois le message SETUP arrivé dans le nœud C, le contrôle d'appel de celui-ci demandera à l'entité de signalisation l'ouverture d'un VC avec la feuille 4 par l'envoi d'un message SETUP UNI. N'oublions pas que ce message transporte une référence de point terminal et un IE Broadband Bearer Capability permettant à la feuille de savoir qu'elle fait partie d'une connexion point-à-multipoint.

Le nœud B est ici un LCN : il dupliquera les données reçues de manière à les envoyer sur la branche menant aux feuilles 1, 2 et 3 ainsi que sur la branche menant à la feuille 4.

4.4 Conclusion

Avec les connaissances que nous avons acquises au chapitre précédent ainsi que dans ce chapitre, nous savons à présent clairement comment se déroulent toutes les procédures de signalisation à partir d'un TE appelant jusqu'à un TE appelé et comment ces procédures sont traitées au sein même d'un réseau. Nous savons quelles sont les informations échangées lors de ces procédures et pourquoi elles sont nécessaires.

Nous avons vu quels étaient les problèmes induits par la nécessité d'une connaissance complète de la topographie d'un réseau nécessaire au routage des messages et comment l'ATM Forum a pu résoudre ce problème par l'introduction du concept de hiérarchie dans les réseaux.

Ce mémoire étant exclusivement consacré à la signalisation, nous avons dû faire un certain nombre d'abstractions lors de l'étude du module de routage du protocole PNNI. Nous n'avons pu, entre autres, aborder en détail le mécanisme d'agrégation des données, la structure des bases de données sur la

topologie, le fonctionnement d'un algorithme de routage devant tenir compte d'une hiérarchie dans les nœuds, les mécanismes internes d'un GCAC et d'un CAC (les paramètres sur lesquels travaillent ces algorithmes, ...) ainsi que les mécanismes précis de fonctionnement des machines Hello, de la synchronisation des bases de données et de la procédure d'inondation. Ces différents sujets mériteraient à eux seuls des études plus approfondies et la rédaction d'un nouveau mémoire.

PNNI suscite un fort engouement sur le marché des télécommunications. Cependant, le protocole n'étant pas encore finalisé et approuvé définitivement par les membres de l'ATM Forum, des produits hardwares ou softwares incluant ce protocole ne devraient pas voir le jour avant le milieu de l'année 1996 au plus tôt, voire début de l'année 1997. Notons toutefois que la société américaine Trillium Digital Systems Inc. commercialise actuellement une couche de signalisation UNI 3.1 et PNNI. Le module de routage PNNI de cette couche ne supporte à l'heure actuelle qu'un réseau *flat-level* - c'est-à-dire sans hiérarchie - et constitué d'un PG unique regroupant tous les nœuds du réseau. La disponibilité de ce produit n'induit pas obligatoirement que celui soit fini : cette société devra plus que vraisemblablement fournir des mises à jour à ses clients suite aux modifications classiques de dernières minutes qui seront introduites dans le protocole PNNI suite à son vote définitif. Le but recherché par cette société est sans nul doute de se positionner en premier sur le marché ouvert par ce protocole.

D'autres sociétés américaines, telles Cisco et Cascade, travaillent actuellement à l'intégration du protocole PNNI dans leurs routeurs ATM.

Dans l'attente de la disponibilité de produits PNNI, des sociétés telle Bellcore commercialise à ce jour des couches de signalisation supportant le protocole PNNI dans sa phase 0, soit IISP.

De nouvelles déclinaisons du protocole PNNI font également leur apparition, tels PNNI Augmented Routing (PAR) et Integrated PNNI (I-PNNI). PAR est une méthode utilisée pour effectuer du routage multiprotocolaire dans un environnement ATM. Ceci consiste à exécuter des protocoles de routage de niveau Internet tels OSPF, IPX, DECnet ou AppleTalk sur ATM, en utilisant PNNI dans les routeurs frontière ATM (i.e. les routeurs avec interface sur le réseau ATM). Ces routeurs frontière ATM ont pour but de se localiser l'un l'autre, démarrer le protocole de routage en "overlay" (OSPF, IPX, ...) et simplifier la maintenance du protocole de routage utilisé (autre que PNNI) sur ATM [PAR96]. I-PNNI est une extension du protocole PNNI utilisée pour le routage IP dans un environnement ATM et a pour but de définir comment le routage doit être accompli dans un tel environnement (comment les SVCs doivent être routés dans un réseau ATM ? Quand et où établir des SVC ?) [IPNNI96].

Le lecteur désireux d'en savoir plus sur PAR ou I-PNNI se mettra en contact avec l'ATM Forum. Les documents de description de PAR et I-PNNI portent respectivement les références *ATM Forum/96-0354* et *ATM Forum/96-0355*.

5. Stage

Ce chapitre traite du stage qui a été effectué au Centre d'Etudes et de Recherches IBM de La Gaude (France) du mois d'août 1995 au mois de décembre 1996 et au centre IBM de Raleigh, Caroline du Nord (USA), au mois de janvier 1996.

Ce chapitre a été classé confidentiel pour des raisons de protection de la propriété industrielle et intellectuelle et ne figure donc pas dans cette version imprimée du mémoire.

6. Conclusion

Nous avons étudié dans ce mémoire la question de la signalisation dans les réseaux ATM privés. Nous avons vu qu'il existe deux cas distincts de signalisation : la signalisation ayant lieu à l'interface entre l'utilisateur et le réseau par l'intermédiaire du protocole UNI 3.1 et celle ayant lieu entre tous les noeuds d'un réseau ATM (ou de plusieurs réseaux) par l'intermédiaire du protocole PNNI. L'étude de ces deux protocoles nous a montré qu'ils étaient tout deux basés sur les mêmes mécanismes d'états associés à une procédure de signalisation et de transitions entre ces états suite à l'envoi ou à la réception de messages particuliers - par ailleurs quasi identiques dans ces deux protocoles - ce qui nous amène à dire que PNNI est en quelque sorte une extension du protocole UNI.

Le protocole le plus intéressant est très certainement le protocole PNNI. Outre les procédures de signalisation classiques, calquées sur le protocole UNI, il introduit des concepts révolutionnaires tels que la construction de la représentation hiérarchique d'un réseau permettant de diminuer de manière remarquable la quantité d'informations concernant la topologie devant être distribuée au sein d'un réseau, la notion de routage partiel à la source combinant les avantages et supprimant les désagréments des routages à la source et de type hop-by-hop ainsi que le mécanisme de crankback permettant au réseau de chercher par lui-même de nouveaux chemins à suivre pour l'ouverture d'une connexion si certains problèmes devaient être rencontrés sur le chemin choisi à l'origine de l'appel. Ce mémoire s'étant concentré principalement sur l'aspect signalisation de ce protocole, nous n'avons pu étudier en détail le module de routage de ce dernier. Toutefois, il mériterait très certainement de faire l'objet d'études plus approfondies.

A la lumière de ce que nous avons vu, nous pouvons donner une définition plus complète d'un protocole de signalisation dans les réseaux ATM privés :

> un protocole de signalisation dans les réseaux ATM privés définit comment une connexion peut être établie, sur demande d'un utilisateur ou d'un noeud du réseau, avec un ou plusieurs utilisateurs ou noeuds distants et comment cette connexion doit être terminée. Plus précisément, ce protocole doit pouvoir supporter un certain nombre de caractéristiques [BLA95] :
>
> - l'établissement et la fermeture de connexions sur demande;
> - le support de connexions point-à-point et point-à-multipoint;
> - l'existence de procédures spécifiques permettant d'effectuer une demande d'ouverture de connexion, l'acceptation ou le refus de celle-ci, la fermeture de la connexion, le redémarrage d'une connexion ainsi que l'ajout et le retrait de feuilles à une connexion point-à-multipoint;
> - la réservation symétrique ou asymétrique de largeur de bande, permettant l'ouverture de connexions bidirectionnelles à largeurs de bande identiques ou différentes;
> - la négociation d'un contrat de trafic avec le réseau mais pas avec l'utilisateur distant;
> - le support des classes de trafic de type A, C et X.
>
> De plus, pour la signalisation au sein même d'un réseau, le protocole doit pouvoir :
>
> - utiliser la construction hiérarchique qui aura été élaborée par le protocole de routage afin de diriger un message de demande d'ouverture de connexion au travers du réseau;

- être capable de prendre les mesures nécessaires afin de rediriger une demande d'ouverture de connexion au travers du réseau si un problème de disponibilité de ressources ou d'accessibilité devait avoir lieu au sein du réseau.

Chacun de ces deux protocoles - UNI et PNNI - demandent l'existence d'un canal virtuel connectant les entités paires de signalisation et permettant le transport des messages relatifs aux procédures de signalisation. Ce canal virtuel doit être sûr, i.e. le transport des données de signalisation par ce canal doit pouvoir être effectué sans erreurs et il doit être possible d'effectuer un contrôle de flux sur les informations circulant au travers de ce canal. Afin d'obtenir ce canal et donc avant toute procédure de signalisation, nous avons vu qu'une entité de signalisation, qu'elle soit UNI ou PNNI, demande à sa couche inférieure, la couche SAAL, l'ouverture d'un tel canal. Cette couche est, comme nous l'avons vu, divisée en deux sous-couches : SSCS et CP. Le rôle de la sous-couche SSCS, subdivisée elle-même en les modules SSCF et SSCOP, est précisément d'ouvrir un tel canal et d'offrir un service de transfert fiable à la couche de signalisation, c'est-à-dire sans erreurs et avec contrôle du flux d'informations.

Toujours dans la couche SAAL, nous avons vu que les informations en provenance de la couche de signalisation et fournies à la sous-couche SSCS sont ensuite délivrées à la deuxième sous-couche de SAAL, la sous-couche CP. Dans cette sous-couche, les informations reçues de la sous-couche SSCS sont découpées en champs de 48 octets qui sont par la suite délivrés à la couche ATM. Celle-ci peut ensuite encapsuler ces champs dans des cellules ATM et les délivrer à la couche physique qui les transmet sur le support physique.

Tout au long du chapitre traitant de l'architecture du modèle de signalisation et plus particulièrement pour l'étude de SSCF et SSCOP, nous avons fait l'hypothèse de l'existence du plan de gestion mais nous n'avons jamais étudié ni détaillé celui-ci. A l'heure actuelle, le plan de gestion est fort probablement l'aspect le moins standardisé de la technologie ATM. Une étude précise et approfondie des rôles qu'un tel plan a à jouer ainsi que des protocoles propriétaires ou ouverts déjà spécifiés constituerait sans nul doute un sujet fort intéressant de futur mémoire.

ACRONYMES

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
AFI	Authority and Format Identifier
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BSNT	Backward Sequence Number to be Transmitted
CAC	Call Admission Control
CBR	Constant Bit Rate
CCITT	Consultative Committee on International Telegraphy and Telephony
CDV	Cell Delay Variation
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CP	Convergence Part
CPA	Control Point Adapter
CPCS	Common Part Convergence Sublayer
CPE	Customer Premises Equipment
CTD	Cell Transfer Delay
DCD	Data Country Code
DFI	Domain Format Identifier
DSP	Domain Specific Part
DTL	Designated Transit List
FSM	Finite State Machine
FSNC	Forward Sequence Number of the last message to be aCcepted by peer
GFC	Generic Flow Control
HDLC	High level Data Link Control
HEC	Header Error Control
HO-DSP	High Order Domain Specific Part
ICD	International Code Designator
ID	Interface Data
IDI	Initial Domain Identifier
IDP	Initial Domain Part
IE	Information Element
IISP	Interim Interswitch Signaling Protocol
ILMI	Interim Local Management Interface
IP	Internet Protocol
I-PNNI	Integrated Private Network-to-Network Interface
ISO	International Standards Organization
LCN	Last Common Node
LIJ	Leaf Initiated Join
ITU	International Telecommunication Union
MBS	Maximum Burst Size
MIB	Management Information Base
MSPE	Meta-Signaling Protocol Entity
MU	Message Unit
NBBS	Network BroadBand System

NNI	Network-to-Node Interface ou Network-to-Network Interface
PAR	PNNI Augmented Routing
PNNI	Private Network-to-Network Interface ou Private Node-to-Network Interface
OAM	Operation Administration Maintenance
OSI	Open System Interconnection
PCM	Pulse Code Modulation
PCR	Peak Cell Rate
PD	Protocol Discriminator
PDU	Protocol Data Unit
PG	Peer Group
PGL	Peer Group Leader
PT	Payload Type
PTSE	PNNI Topology State Element
PTSP	PNNI Topology State Packet
PVC	Permanent Virtual Channel
QoS	Quality of Service
RN	Retrieval Number
S-AAL	Signalling ATM Adaptation Layer
SAP	Service Access Point
SAR	Segmentation And Reassembly
SCR	Sustainable Cell Rate
SDU	Service Data Unit
SEL	Selecteur
SN	Sequence Number
SNMP	Simple Network Management Protocol
SSCF	Service Specific Coordination Function
SSCOP	Service Specific Connection Oriented Protocol
SSCOP-UU	SSCOP User-to-User information
SSCS	Service Specific Convergence Sublayer
SVC	Switched Virtual Channel
TDM	Time-Division Multiplexing
TE	Terminal Equipment
UBR	User Bit Rate
UNI	User-to-Network Interface
UPC	User Parameter Control
VBR	Variable Bit Rate
VC	Virtual Channel
VCC	Virtual Channel Connexion
VCI	Virtual Channel Identifier
VP	Virtual Path
VPC	Virtual Path Connexion
VPI	Virtual Path Identifier
WAN	Wide Area Network

BIBLIOGRAPHIE

[ALLES95] Anthony ALLES, *ATM Internetworking*, Cisco Systems, 1995

[BLA95] Uyless BLACK, *ATM : Foundation for broadband networks*, Prentice Hall Series In Advanced Communications Technologies, 1995

[FELD95] Robert Feldman, *Never Say Never To Atm*, Open Computing Magazine Volume 12, N° 3, mars 1995

[FORE95] Fore Systems, Inc., *SPANS NNI : Simple Protocol for ATM Network Signaling at the Network-to-Network Interface*, disponible chez FORE SYSTEMS, 174 Thorn Hill Road Warrendaele, PA 15086, 1995

[IISP94] *Interim Inter-switch Signaling Protocol (IISP) Specification Version 1.0*, ATM Forum af-pnni-0026.000, 1994

[IPNNI96] *Issues and Approaches for Integrated PNNI*, ATM Forum 96-0355, ATM Forum, 1996

[KYAS95] Othmar KYAS, *ATM Networks*, Thomson International Publishing, 1995

[NSAP91] Richard COLELLA, Ella GARDNER, Ross CALLO N, *Guidelines for OSI NSAP Allocation in the Internet (RFC1237)*, Network Working Group, 1991

[PAR96] *Overview of PNNI Augmented Routing*, ATM Forum 96-0354, ATM Forum, 1996

[PNNI94] *Private Network-to-Network Interface Draft Specification*, ATM Forum 94-0471R13, ATM Forum, 1994

[Q.2110-94] *Q.2110 : B-ISDN ATM Adaptation Layer - Service Specific Connection Oriented protocol (SSCOP)*, ITU-T, 1994

[Q.2120-94] Study Group 11, *Q.2120 : B-ISDN Meta-Signalling Protocol*, ITU-T, 1994

[Q.2140-94] Study Group 11, *Q.2140 : B-ISDN Signalling ATM Adaptation Layer - Service Specific Coordination Function for Support of Signalling at the Network Node Interface (SSCF at NNI)*, ITU, 1994

[STA92] William STALLINGS, *ISDN and Broadband ISDN*, second edition, Macmillan Publishing Company, 1992

[TRAF4-96] *Trafic Management Specification Version 4.0*, ATM Forum af-95-0013R11, ATM Forum, Mars 1996

[UNI3.1-94] *ATM User-Network Interface Specification Version 3.1*, ATM Forum, 1994

[UNI4.0-95] *ATM User-to-Network Interface Signaling Specification Version 4.0 Release 11*, ATM Forum, 1995

Internet

Documents HTML :

[ATMHOME] *ATM Home* : http://ganges.cs.tcd.ie:80/4ba2/atm/index.html

[CELLR] *The Cell Relay Retreat* : http://cell-relay.indiana.edu

[CHIROUZE] *Michel Chirouze* : http://www.lirmm.fr/atm/

[FORUM] *ATM Forum Home Page* : http://www.atmforum.com

[NORM] *Norm Al Dude and Professor N. Erd on the subject of ATM* : http://www.datacomm-us.com/technow/scan06/scan06.html

[SREDDIVA] *Asynchronous Transfer Mode (ATM)* : http://cne.gmu.edu/~sreddiva/Texttut.html

[VIVID] Joe M. HALPERN, *The Architecture and Status of PNNI*, Newbridge Network Inc., http://www.vivid.newbridge.com/documents/Joel.html

Liste de discussion - Newsgroup :

Cell Relay : pour renseignements, envoyer un e-mail à comp.dcom.cell-relay@indiana.edu

ANNEXES

Annexe A : Détection et correction d'erreurs avec HEC

Le champ HEC de la cellule ATM est utilisé afin d'effectuer un contrôle d'erreurs sur l'en-tête de la cellule. Un champ HEC d'une taille de 8 bits a été choisi car il permet la détection et la correction d'une erreur d'un bit dans l'en-tête ou la détection de plusieurs bits erronés (mais pas leur correction) [BLACK95].

La logique générale du mécanisme utilisé pour la correction et détection d'erreurs est donné à la Figure A-1.

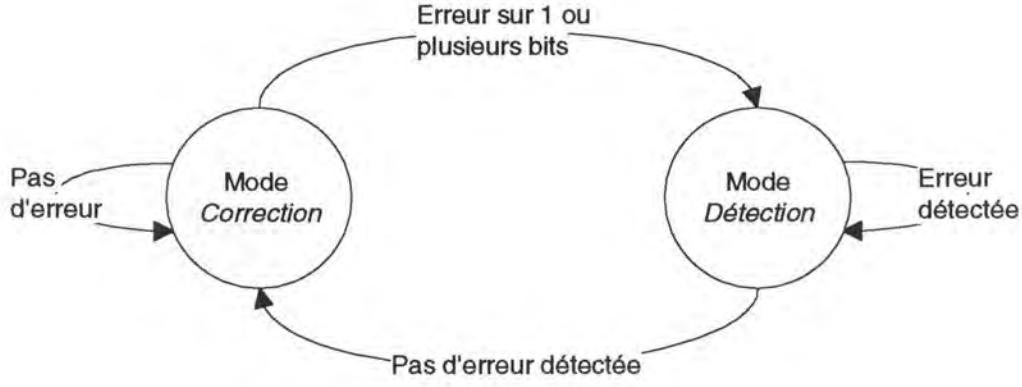


Figure A-1 : correction et détection d'erreurs avec le HEC

Le mécanisme de correction et détection d'erreurs démarre en mode « correction ». Si l'on détecte une erreur portant sur un ou plusieurs bits de l'en-tête, on passe dans le mode « détection ». Si l'erreur porte uniquement sur un bit, l'erreur est corrigée, sinon la cellule est effacée et donc non délivrée aux couches supérieures. Tant que l'on se trouve dans le mode « détection », toute cellule reçue comportant des erreurs, qu'elles portent sur un ou plusieurs bits, est effacée. On ne repasse dans le mode « correction » que suite à la réception d'une cellule dont l'en-tête n'est pas erroné.

Annexe B : formats des PDU SSCOP

Type d'opération générée	*Nom du PDU*	*Champ type de PDU*	*Description*
Etablissement de connexion	BGN	0001	Demande de connexion
	BGAK	0010	Accord de connexion
	BGREJ	0111	Rejet de connexion
Fermeture de connexion	END	0011	Demande de fermeture
	ENDAK	0100	Acceptation de la fermeture
Resynchronisation	RS	0101	Demande de resynchronisation
	RSAK	0110	Acceptation de la resynchronisation
Transfert garanti de données	SD	1000	Demande d'envoi de données en mode garanti
	POLL	1010	Etat de l'entité émettrice + demande de l'état de l'entité réceptrice
	STAT	1011	Etat de l'entité réceptrice suite à une demande de l'entité émettrice
	USTAT	1100	Etat de l'entité réceptrice suite à aucune demande
Transfert non garanti de données	UD	1101	Envoi de données en mode non garanti
Transfert de données de gestion	MD	1110	Envoi de données de gestion

Tableau B-1 : liste des PDU SSCOP

Les primitives à l'interface SSCOP/SSCF génèrent un ensemble de 15 PDU différents, répartis suivant le type d'opération générée, comme l'indique le Tableau B-1.

1) Etablissement de connexion

I- Format

Les PDU impliqués lors de l'ouverture d'une connexion SSCOP sont exposés à la Figure B-1.

Le champ PL utilisé dans ces PDU spécifie le nombre d'octets de padding qu'il a fallu ajouter. Sa valeur ne peut sortir de l'intervalle [0;3] octets. Dans le cas où les PDU BGN, BGAK et BGREJ ne contiennent aucune information utilisateur SSCOP-UU, le champ de padding est mis à zéro.

Le champ N(MR) spécifie le numéro de séquence du premier SD PDU non accepté par l'utilisateur. Si l'entité SSCOP réceptrice reçoit un PDU dont le numéro de séquence est plus élevé que N(MR), ce SD PDU est effacé et l'entité réceptrice enverra un STAT PDU à l'entité émettrice.

Le champ N(SQ) identifie le numéro de séquence de la connexion. Lors de l'envoi du premier BGN PDU, le numéro N(SQ) est sauvé dans une variable locale VR(SQ). Par la suite, lors de la réception d'un BGN PDU, la comparaison de N(SQ) et VR(SQ) permet de voir s'il s'agit d'un PDU retransmis ou non.

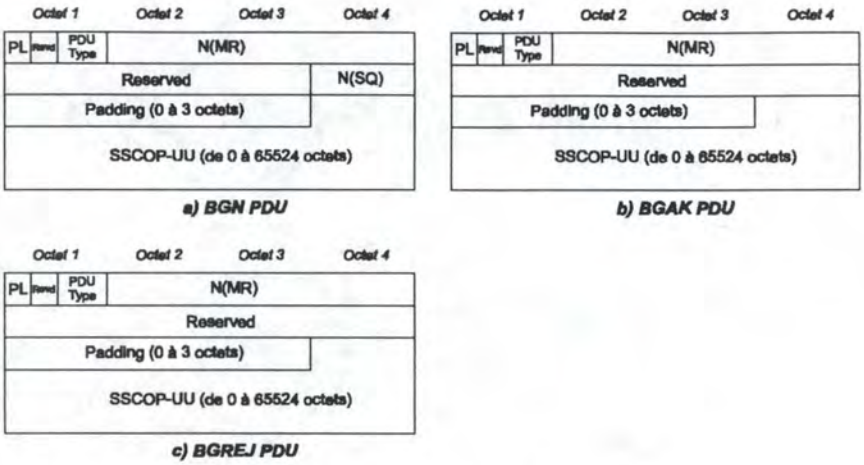


Figure B-1 : PDU SSCOP pour l'ouverture d'une connexion

II- Scénario

La génération d'un AA-ESTABLISH.request entraîne l'envoi d'un BGN PDU vers l'entité réceptrice. La réception de ce PDU génère un AA-ESTABLISH.indication à l'utilisateur de l'entité SSCOP réceptrice. Si la connexion est acceptée, la génération d'un AA-ESTABLISH.response enverra un BGAK PDU à l'entité émettrice, générant un AA-ESTABLISH.confirmation. Si la connexion est refusée, la génération d'un AA-RELEASE.request enverra un BGREJ PDU à l'entité émettrice.

2) Fermeture de connexion

I- Format

La Figure B-2 reprend le format des PDU SSCOP utilisés lors de la fermeture d'une connexion.

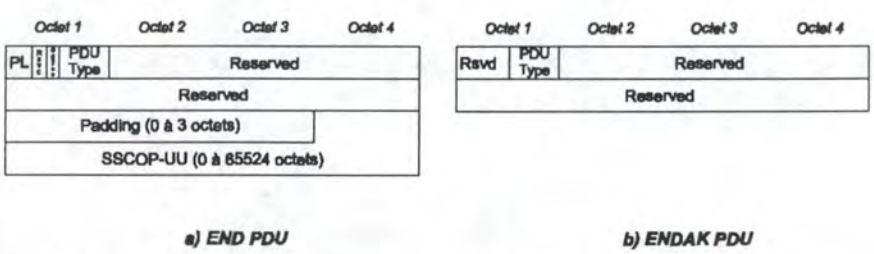


Figure B-2 : PDU SSCOP pour la fermeture de connexion

II- Scénario

La génération d'un AA-RELEASE.req entraîne l'envoi d'un END PDU vers l'entité réceptrice. La réception de ce PDU génère un AA-RELEASE.indication à l'utilisateur de l'entité SSCOP réceptrice. La réception du END PDU génère automatiquement l'envoi d'un ENDAK PDU, déclenchant un AA-RELEASE.confirmation à l'utilisateur de l'entité SSCOP émettrice.

3) Resynchronisation

I- Format

La Figure B-3 reprend le format des PDU SSCOP utilisés lors de la procédure de resynchronisation.

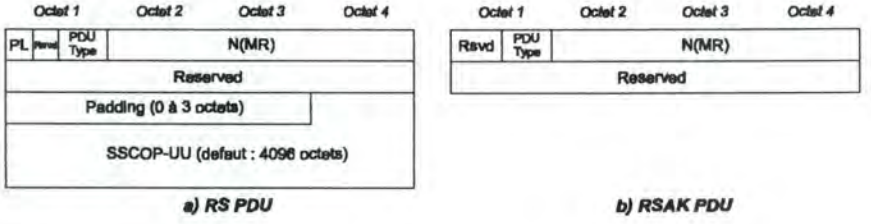


Figure B-3 : PDU SSCOP pour la procédure de synchronisation

II- Scénario

La génération d'un AA-RESYNC.request entraîne l'envoi d'un RS PDU vers l'entité réceptrice. La réception de ce PDU génère un AA-RESYNC.indication à l'utilisateur de l'entité SSCOP réceptrice. En réponse à cette demande, l'utilisateur génère un AA-RESYNC.response qui enverra un RSAK PDU à l'entité émettrice, générant un AA-RESYNC.confirmation auprès de l'utilisateur de l'entité SSCOP.

4) Transfert de données garanti

I- Format

La Figure B-4 reprend le format des PDU SSCOP utilisés lors du transfert d'informations en mode garanti.

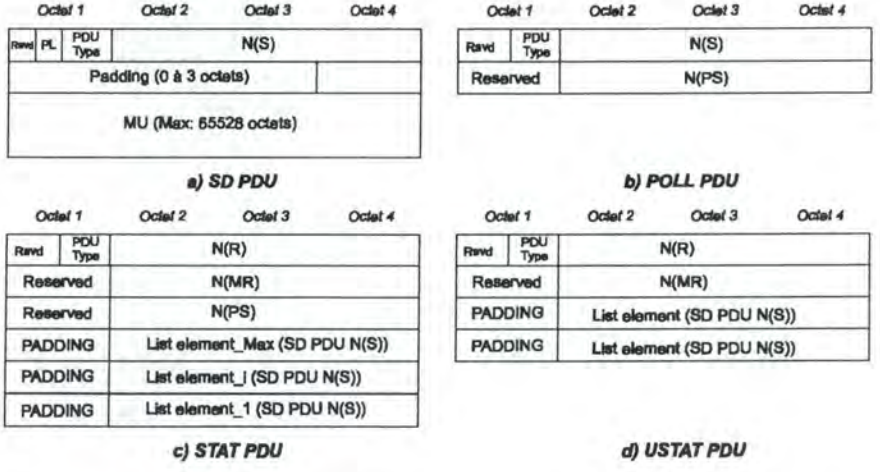


Figure B-4 : PDU SSCOP impliqués dans le transfert garanti de données

Le SD PDU (Figure B-4 a) est utilisé pour le transport de PDU numérotés séquentiellement (numérotation avec le numéro de séquence N(S)) dont les champs d'information proviennent de l'utilisateur SSCOP.

Le POLL PDU est utilisé pour demander des informations de statut concernant l'entité paire. Le champ N(PS) transporté dans le POLL PDU est le numéro de séquence POLL de ce message. Il est incrémenté avant chaque envoi d'un POLL PDU.

Le STAT PDU est défini comme réponse au POLL PDU. Il renferme des informations quant à l'état de réception de l'entité, ainsi que le numéro de séquence N(PS) permettant de différencier des STAT PDU répondant à des POLL PDU différents. Le STAT PDU transporte également le numéro de séquence N(R) du prochain SD PDU attendu.

Le USTAT PDU est utilisé lorsqu'une entité SSCOP a remarqué qu'elle n'avait pas reçu certains SD PDU. Elle peut détecter ceci en se basant sur le numéro de séquence du SD PDU.

II- Scénario

L'envoi d'un message SD PDU se fait suite à l'utilisation de la primitive AA-DATA.request. Plutôt que de s'attarder sur la description des séquences de primitives et messages qui sont assez triviales, nous exposons ici l'utilisation de messages STAT, POLL et USTAT suite à la détection de SD PDU manquants.

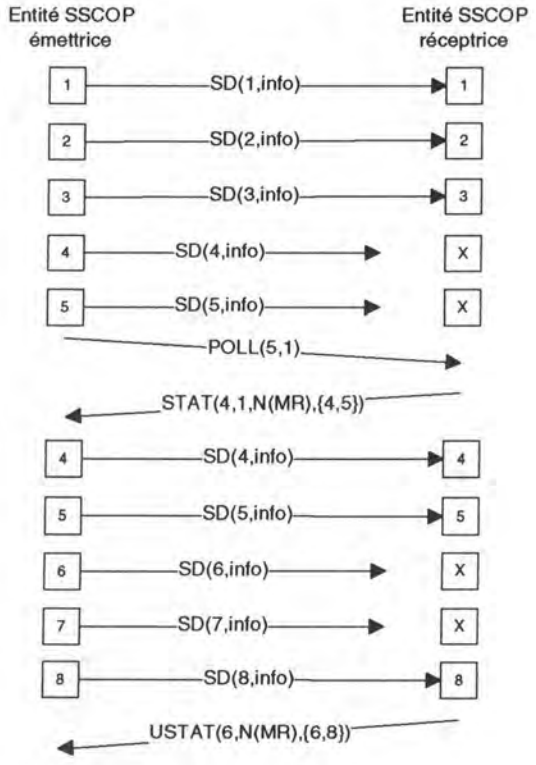


Figure B-5 : détection et correction d'erreurs dans SSCOP

Dans la Figure B-5, l'entité SSCOP émettrice doit envoyer les PDU numérotés 1 à 8 à l'entité SSCOP réceptrice. L'envoi de ces données se fait en mode garanti, en utilisant donc les SD PDU dont la structure est donnée à la Figure B-4.

L'entité SSCOP réceptrice reçoit bien les SD PDU numérotés de 1 à 3, mais perd les PDU 4 et 5. Elle ne peut s'en rendre compte, n'ayant pas encore reçu de SD PDU ayant un numéro de séquence qui pourrait lui indiquer cette perte. Cependant, après avoir envoyé ses 5 premiers SD PDU, l'entité émettrice demande un accusé de réception sur ces PDU par l'intermédiaire d'un POLL PDU. Les

paramètres passés dans ce PDU sont : N(S) et N(PS), où N(S) est le numéro de séquence du dernier SD PDU envoyé et N(PS) est le numéro de séquence POLL de ce PDU (1 car le premier POLL PDU).

L'entité réceptrice remarque alors, par le numéro N(S) fourni dans le POLL PDU, qu'elle a perdu les SD PDU numérotés 4 et 5. Elle le signale à l'entité émettrice par l'intermédiaire d'un STAT PDU, dont les paramètres sont : N(R), N(MR), N(PS) et list, où N(R)=4 (le prochain SD PDU attendu), N(PS)=1 (réponse au POLL PDU de séquence 1), et list = {4,5} afin de signaler la perte des SD PDU 4 et 5. Ces SD PDU sont renvoyés par l'entité émettrice, qui continuera alors à envoyer les SD PDU restants.

Durant la réception des SD PDU de séquence 4 à 8, l'entité réceptrice perd les SD PDU 6 et 7, puis reçoit le SD PDU 8. Le dernier numéro de séquence reçu étant 5, l'entité réceptrice remarque la perte des SD PDU 6 et 7. Elle en avertit l'entité émettrice par l'intermédiaire d'un USTAT PDU, dont les paramètres sont : N(R), N(MR) et list, où N(R)=6 (le SD PDU attendu après 5), et list = {6,8} afin de demander la retransmission des SD PDU de séquence 6 à 8.

5) Transfert de données non garanti et transfert avec la gestion des couches

I- Format

La Figure B-6 reprend le format des PDU utilisés pour le transfert des données en mode non garanti et pour le transfert de données avec la gestion des couches.

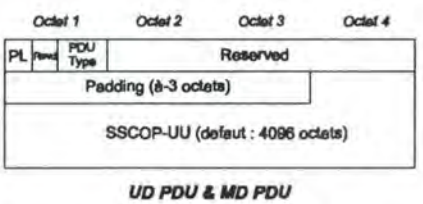


Figure B-6 : SSCOP PDU impliqués dans le transfert non garanti de données et de management

On remarquera que ce PDU ne transporte pas de numéro de séquence : le mode de transfert étant non garanti, il n'existe pas de mécanisme afin d'assurer un respect de séquence et une détection de PDU manquant.

II- Scénario

L'UD PDU (Unacknowledged Data Transfer PDU) est utilisé pour le transfert de données non garanti entre deux utilisateurs SSCOP. Il est envoyé suite à la génération d'un AA-UNITDATA.request auprès de l'entité SSCOP émettrice. Sa réception (pour peu que le PDU ne soit pas perdu) génère un AA-UNITDATA.indication auprès de l'utilisateur SSCOP récepteur.

Le MD PDU (Management Data PDU) est utilisé pour le transfert d'informations entre entités SSCOP et entités de gestion des couches paires. L'envoi d'un MD PDU se fait suite à la génération d'un MAA-UNITDATA.request auprès de l'entité SSCOP émettrice. La réception de ce PDU génère un MAA-UNIDATA.indication auprès de l'entité de gestion des couches paires.

Annexe C : Operation - Administration - Maintenance

Le trafic de cellules OAM (OAM : Operation Administration Maintenance) est utilisé dans un but de gestion et de maintenance du réseau. La gestion et la maintenance se rapportent exclusivement à la couche physique et à la couche ATM. Cinq niveaux de hiérarchie ont été définis, chacun d'eux se rapportant à des caractéristiques précises d'une connexion sur le réseau :

- Niveau F5 : le niveau F5 se situe dans la couche ATM. Le trafic OAM se passe entre entités concernées par les connexions de type VC. Il traite des dégradations de performances des VC, des pertes ou des retards dans l'arrivée des cellules, ainsi que des problèmes d'insertion de cellules.
- Niveau F4 : le niveau F4 se situe dans la couche ATM. Le trafic OAM se passe entre entités concernées par les connexions de type VP. Il traite principalement des problèmes de disponibilité des VP.
- Niveau F3 : le niveau F3 se situe dans la couche physique. Le trafic OAM se passe entre éléments s'occupant du HEC, du cell delineation et de l'assemblage/désassemblage du payload et de l'en-tête des cellules.
- Niveau F2 : le niveau F2 se situe dans la couche physique. Le trafic OAM se passe entre fins de sections digitales (le lien reliant directement deux ressources ATM entre elles). Il traite des problèmes de synchronisation des trames de transmission et des dégradations de performances.
- Niveau F1 : le niveau F1 se situe dans la couche physique. Le trafic OAM transporte des informations de gestion et maintenance à propos de problèmes pouvant survenir entre sections digitales se situant de part et d'autre d'un régénérateur; il s'agit principalement de problèmes de synchronisation et de perte des trames de transmission.

On parle de trafic de cellules OAM et non de messages OAM car les informations transportées tiennent toutes dans une seule cellule.

La structure des cellules OAM de niveau F4 et F5 est donnée à titre d'exemple à la Figure C-1.

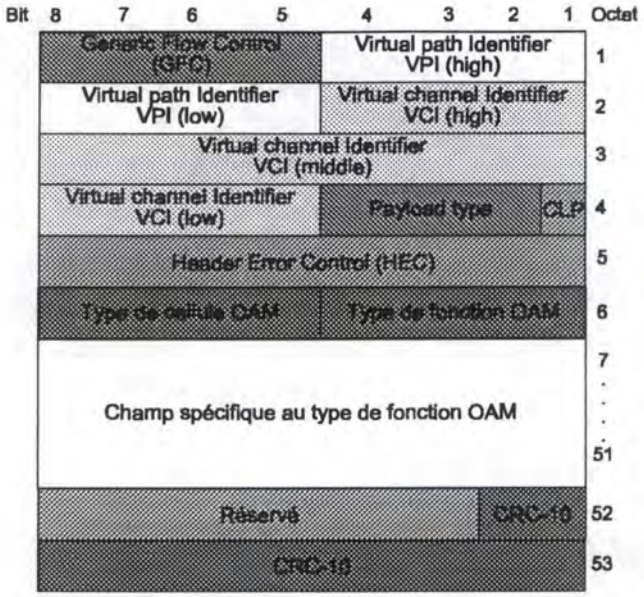


Figure C-1 : cellules OAM pour les niveaux F4 et F5

Dans cette figure, les champ VPI, VCI et PTI (Payload Type Identifier) prennent les valeurs suivantes :

- *pour le niveau F5* : les champs VPI et VCI prennent la même valeur que le couple (VPI, VCI) identifiant le VC à propos duquel on veut échanger des informations de gestion et de maintenance. Le champ PTI indique si l'on désire échanger des informations de gestion et maintenance de bout en bout (i.e. entre les 2 TE connectées) ou uniquement sur un segment (i.e. entre deux commutateurs ou entre un commutateurs et un TE qui y serait connecté).
- *pour le niveau F4* : le champ VPI prend la même valeur que celui des cellules utilisateur empruntant le VP concerné. Le champ VCI indique si l'on désire échanger des informations de gestion et maintenance de bout en bout ou uniquement sur un segment. Le champ PTI n'a pas de signification particulière.

Annexe D : Information Elements

1) Identifiant d'un IE

Le Tableau D-1 reprend l'ensemble des identifiants des IE principaux que l'on peut trouver dans un message UNI. Une très brève description de ces IE y est également donné. Nous présenterons par la suite quelques IE en détail.

IE	*Codage binaire de l'identifiant*	*Rôle*
Cause	00001000	Permet de spécifier pourquoi certains messages tels que RELEASE ou STATUS ont été générés
Etat de la procédure de connexion	00010100	Décrit l'état actuel d'une procédure de connexion ou encore l'état de la procédure de redémarrage
Référence de point terminal	01010100	Permet d'identifier une feuille particulière faisant partie d'une connexion point-à-multipoint
Etat du point terminal	01010101	Permet d'indiquer l'état d'une feuille prenant part à une connexion point-à-multipoint
Paramètres AAL	01011000	Indique les paramètres caractérisant le type d'AAL qui va être utilisé après complétude de la procédure de signalisation
Descripteur de trafic ATM	01011001	Indique les caractéristiques du trafic que l'on va générer (PCR, MBS, SCR)
Identificateur de connexion	01011010	Permet d'identifier les ressources locales à l'interface TE / point d'accès (VPI/VCI)
Qualité de service	01011100	Permet de demander une classe de QoS lors d'une demande d'ouverture de connexion
Adresse du TE appelant	01101100	Identifie le TE appelant
Sous-adresse du TE appelant		Permet de transporter à travers un réseau public ne supportant que le format public E.164 l'adresse ATM privée du TE appelant
Adresse du TE appelé	01110000	Identifie le TE appelé
Sous-adresse du TE appelé	01110001	Permet de transporter à travers un réseau public ne supportant que le format public E.164 l'adresse ATM privée du TE appelé
Indicateur de redémarrage	01111001	Permet de spécifier une procédure de connexion particulière ou toutes les procédures à faire redémarrer
Broadband Bearer Capability	01011110	Permet de demander le type de service que l'on attend du réseau (point-à-point, point-à-multipoint, CBR, VBR, synchronisation temporelle)

Tableau D-1 : liste des identifiants de IE principaux

2) Structure des IE

Nous présentons dans cette section la structure complète des IE les plus couramment utilisés dans les messages UNI.

I- Cause

Le rôle de l'IE Cause a été donné au Tableau D-1. Cet IE est illustré à la Figure D-1.

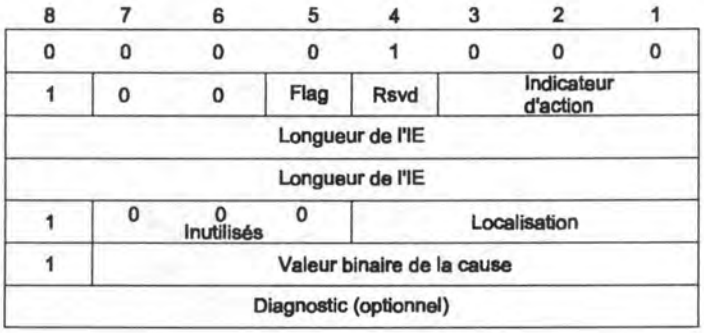


Figure D-1 : IE Cause

Les 4 premiers octets de l'IE Cause composent l'en-tête de celui-ci. La description de l'en-tête a été donnée au chapitre 3, section "Structure d'un message UNI".

Nous reprenons par la suite les éléments composant la structure propre de l'IE Cause (à partir du 5ème octet).

Localisation

Les 4 bits du champ localisation permettent de préciser l'endroit où s'est produite une erreur (si l'IE Cause est transporté dans un message servant à signaler une erreur, tel le message STATUS) et pour un message de type RELEASE, la localisation précise qui a demandé la fermeture de la connexion (le TE ou un point d'accès au réseau). Pour un message STATUS destiné à donner l'état d'une entité paire, la localisation indiquera s'il s'agit de l'état d'un ATM TE ou de l'état d'un point d'accès au réseau. Le codage donné au Tableau D-2 est utilisé pour le champ localisation :

Codage	*Localisation*
0000	Utilisateur (ATM TE)
0001	Réseau privé servant l'utilisateur local (le point d'accès auquel est connecté le TE recevant cet IE)
0010	Réseau public servant l'utilisateur local
0011	Réseau de transit
0100	Réseau public servant l'utilisateur distant
0101	Réseau privé servant l'utilisateur distant
0111	Réseau international
1010	Réseau au delà du point d'interconnexion entre réseau

Tableau D-2 : codage du champ localisation

Valeur de la cause

Ce champ est utilisé afin de spécifier la valeur binaire permettant d'identifier une cause. Toutes les causes encodables dans ce champ sont données au Tableau D-3. Les causes y ont été classées par type.

La colonne "Diagnostic" de ce tableau est expliquée à la section suivante.

Codage	*Signification*	*Diagnostic*
	Evénement Normal	
000 0001	Numéro/Adresse non alloué(e)	Condition permanente/temporaire, normale/anormale
000 0010	Pas de route vers le réseau de transit spécifié	
000 0011	Pas de route vers la destination	Condition permanente/temporaire, normale/anormale
001 0000	Fin de connexion normale	
001 0001	Utilisateur (ATM TE appelé) occupé	
001 0010	L'utilisateur (ATM TE appelé) ne répond pas	
001 0101	Appel rejeté	Raison du rejet
001 0110	Le numéro/l'adresse a changé(e)	Nouvelle adresse
001 0111	Le TE rejette tous les appels	
001 1011	Destination en panne	
001 1100	Format d'adresse invalide (ou incomplet)	
001 1110	Réponse à un message STATUS ENQUIRY	
001 1111	Normal/non spécifié (pas de condition d'erreur)	
	Ressource non disponible	
010 0011	VPI/VCI demandé non utilisable	
010 0100	Impossibilité d'assigner le couple VPI/VCI demandé	
010 0101	PCR, SCR spécifié non disponible	Identification du champ du descripteur de trafic
010 0110	Réseau hors service	
010 1001	Panne temporaire	
010 1011	Suppression d'un IE reçu	Identifiant de l'IE ignoré (ou plusieurs IE)
010 1101	Pas de VPI/VCI disponible	
010 1111	Ressource non disponible (non spécifié)	
	Service ou option non disponible	
011 0001	QoS non disponible	Condition permanente/temporaire, normale/anormale
011 1001	Broadband Bearer Capability non autorisée	
011 1010	Broadband Bearer Capability non disponible	
011 1111	Service ou option non disponible (non spécifié)	
	Service ou option non implémenté(e)	
100 0001	Broadband Bearer Capability non implémentée	
100 1001	Combinaison non supportée de paramètres descripteurs de trafic	
100 1110	Paramètres AAL non supportés	
	Service ou option non implémenté(e)	
101 0001	Référence d'appel non valide	
101 0010	Canal identifié non existant	Couple VPI/VCI non existant
101 1000	Destination incompatible	Identifiant de l'IE faisant défaut (ou plusieurs IE)
101 1001	Référence de point terminal non valide	
101 1011	Sélection du réseau de transit non valide	
101 1100	Trop de demande ADD PARTY en cours	
	Erreur de protocole	
110 0000	IE obligatoire manquant	Identifiant de l'IE faisant défaut (ou plusieurs IE)
110 0001	Type de message non existant ou non implémenté	Identifiant du message reçu
110 0011	IE non existant ou non implémenté	Identifiant du message reçu
110 0100	Contenu de l'IE invalide	Identifiant de l'IE faisant défaut (ou plusieurs IE)
110 0101	Message non compatible avec l'état actuel de la procédure	Identifiant du message reçu
110 0110	Envoi du message suite à l'expiration d'un timer	Identification du timer
110 1000	Longueur incorrecte du message	
110 1111	Erreur de protocole (non spécifié)	

Tableau D-3 : valeur du champ cause dans l'IE Cause

Remarquons que ce tableau comporte des identifications de cause qui font appel à des notions de routage ("pas de route vers la destination", "pas de route vers le réseau de transit spécifié"). Le protocole UNI ne traite et n'a connaissance d'aucune notion de routage. Cependant le lecteur pourra constater dans le chapitre de ce mémoire consacré au protocole PNNI que ce dernier doit supporter le

protocole UNI. PNNI est, de plus, un protocole constitué d'une partie de routage et d'une partie de signalisation. Nous verrons entre autres que si le point d'accès au réseau - faisant tourner à la fois le protocole UNI pour la signalisation avec le TE et le protocole PNNI pour la signalisation entre commutateurs - ne peut trouver de route vers la destination spécifiée dans le message SETUP, il renverra alors au TE un message indiquant qu'il n'a pu trouver de route. L'entité de signalisation du TE n'entreprendra aucune action particulière suite à la réception d'un message contenant un IE Cause spécifiant que la route n'a pu être trouvée, mais pourra indiquer cette cause à l'utilisateur ayant demandé l'ouverture de connexion.

Diagnostic

Le champ diagnostic permet de donner plus de renseignements quant à certaines causes. Toutes les causes ne bénéficient pas de ce raffinement.

Le champ diagnostic n'est pas limité à un seul octet, comme représenté à la Figure D-1. Cependant, à cause de la variation des formats de cet élément, nous l'avons représenté à cette figure sur un seul octet.

Reprenons l'une ou l'autre cause et précisons la structure et le contenu exact du champ diagnostic associé :

- Cause : Le canal identifié n'existe pas (101 0010)
  La structure du diagnostic associé à cette cause est donné à la Figure D-2. Elle permet de spécifier le couple VPI/VCI que l'entité de signalisation n'a pu identifier.

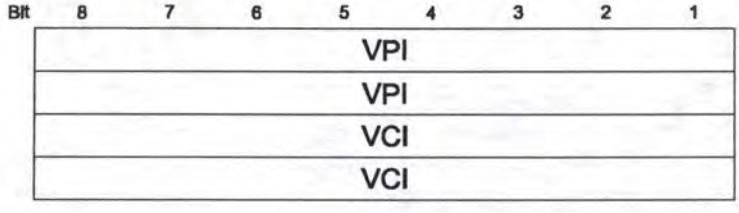


Figure D-2 : champ diagnostic pour la cause 101 0010

- Cause : Appel rejeté (001 0101)
  La structure du diagnostic associé à cette cause est donnée à la Figure D-3. La raison du rejet permet de spécifier si le rejet est propre à l'utilisateur, s'il est dû au manque d'un IE dans le message ou si le contenu d'un IE reçu est insuffisant. La condition informe l'entité réceptrice de cet IE si la cause du rejet est permanente ou transitoire.
  Si la raison du rejet est due au manque d'un IE ou à un IE non complet, on mettra dans le champ "Identifiant d'un IE" l'identifiant de l'IE faisant défaut. Le champ "Diagnostic spécifique à l'utilisateur" ne sera alors pas représenté dans le message (son octet n'existe pas).
  Si la raison du rejet est due à l'utilisateur, celui-ci encode sa raison dans le champ "Diagnostic spécifique à l'utilisateur" selon une syntaxe qui lui est propre.

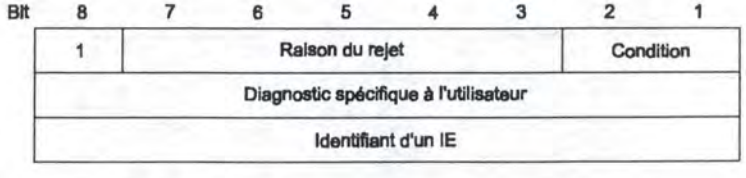


Figure D-3 : champ diagnostic pour la cause 001 0101

II- Etat de la procédure de connexion (Call State)

Le rôle de l'IE Call State est de transporter l'état de la procédure de connexion de l'entité paire (voir chapitre 3, section "Messages pour les procédures de demande d'information"). Cet IE est illustré Figure D-4.

Figure D-4 : IE Call State

Le champ "Etat de la connexion / Etat pour référence globale" est encodé comme selon le Tableau D-3.

Codage	*Etat*
	Etats pour procédure de connexion
00 0000	U0. N0
00 0001	U1. N1
00 0011	U3. N3
00 0110	U6. N6
00 1000	U8. N8
00 1001	U9. N9
00 1010	U10. N10
00 1011	U11. N11
00 1100	U12. N12
	Etats associés à la référence globale
00 0000	Rest0
11 1101	Rest1
11 1110	Rest2

Figure D-5 : codage des états dans l'IE Call State

III- Numéro/Adresse du TE appelé (Called Party Number)

L'IE "Adresse du TE appelé" est utilisé dans le message SETUP. Il permet de spécifier l'adresse du TE avec lequel on désire établir une connexion. Sa structure est représentée à la Figure D-6.

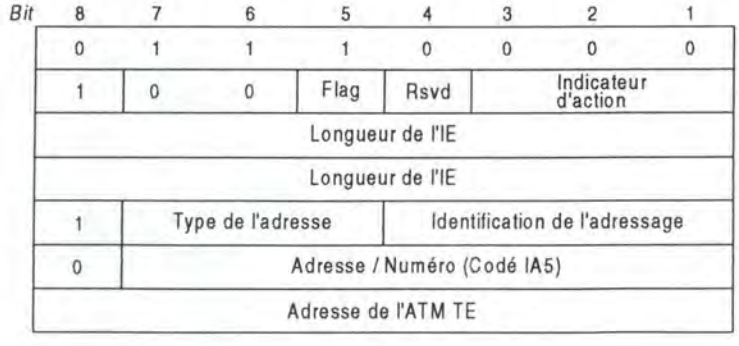


Figure D-6 : structure de l'IE "Adresse du TE appelé"

Reprenons les divers champs que l'on peut trouver dans cette structure :

- Type de l'adresse : l'adresse suit un format international (codé 001) ou utilise un format non connu (codé 000).
- Identification de l'adressage : soit l'adresse donnée est du type E.164 public (codé 0001). Dans ce cas, le type d'adresse sera "international". Soit l'adresse suit le format ATM privé (codé 0010). Dans ce cas, le type d'adresse sera "non connu".
- Adresse / Numéro : dans le cas où l'adresse ATM est un numéro E.164 public elle sera encodée à partir du 6ème octet dans l'ordre des numéros de cette adresse (tel qu'on le taperait sur un clavier téléphonique) avec le 8ème bit toujours à zéro. Les chiffres du numéro sont encodés en caractères IA5.
- Adresse du TE appelé : dans le cas où l'adresse ATM est une adresse privée, elle sera encodée à partir du 6ème octet dans le format défini par ISO 8348/AD 2.

IV- Identifiant de connexion (Connection Identifier)

L'IE "Identifiant de connexion" permet d'identifier les ressources locales (à l'interface UNI) utilisées pour la connexion. Sa structure est donnée à la Figure D-7.

Les champs "Signalisation VP associée" et "Préférée/Exclusive" ne sont pas utilisés dans le protocole UNI, version 3.1. Cependant, on se reportera à l'annexe H pour une description de leur fonction dans le protocole PNNI.

Bit	8	7	6	5	4	3	2	1
	0	1	0	1	1	0	1	0
	1	0	0	Flag	Rsvd	Indicateur d'action		
	Longueur de l'IE							
	Longueur de l'IE							
	1	0 0 Inutilisés		Signalisation VP associée		Préférée / Exclusive		
	VPI							
	VPI							
	VCI							
	VCI							

Figure D-7 : structure de l'IE "Identifiant de connexion"

V- Broadband Bearer Capability

La rôle de cet IE a été donné au chapitre 3 lors de la définition du message SETUP. La description de l'IE Broadband Bearer Capability est donnée à la Figure D-8.

Bit	8	7	6	5	4	3	2	1
	0	1	0	1	1	1	1	0
	1	0	0	Flag	Rsvd	Indicateur d'action		
	Longueur de l'IE							
	Longueur de l'IE							
	1	0 0 Inutilisés		Bearer Class				
	1	0 0 Inutilisés		Type de trafic			Indicateur de synchronisation	
	1	Susceptibilité au clipping		0 0 0 Inutilisés			Configuration de la connexion	

Figure D-8 : structure de l'IE Broadband Bearer Capability

Reprenons les différents champs constituant cet IE :

- Le Bearer Class : le bearer class est utilisé pour les cas d'interworking. Il existe trois types de bearer class : BCOB-A (codé 00001), BCOB-C (codé 00011) et BCOB-X (codé 10000). Si BCOB-A ou BCOB-C sont spécifiés, cela signifie que l'utilisateur demande plus qu'un service ATM simple (i.e. il y aura de l'interworking) et le noeud de réseau concerné par cet interworking doit obligatoirement regarder l'IE contenant les paramètres AAL afin de pouvoir exécuter cet interworking. Si BCOB-X est spécifié, il n'y aura pas d'interworking et aucun noeud n'aura à se soucier des paramètres AAL.
- Le type de trafic permet de spécifier le type de trafic que l'on va générer : CBR (codé 001) , VBR (codé 010) ou non spécifié (codé 000).
- L'indicateur de synchronisation permet de spécifier si la synchronisation temporelle entre les deux ATM TE est nécessaire (codé 01) ou non (codé 10).
- La configuration de la connexion permet de spécifier le mode de connexion que l'utilisateur (le TE) désire : point-à-point (codé 00) ou point-à-multipoint (codé 01).
- La susceptibilité au *clipping* : lorsqu'une demande de connexion part du TE appelant vers le TE appelé, la largeur de bande nécessaire pour supporter le trafic décrit dans le descripteur de trafic est réservée mais non "engagée" ou directement mise à l'utilisation tout au long du chemin traversé par la demande de connexion. Ce n'est qu'une fois que le TE appelé répond par un message CONNECT que la largeur de bande réservée est engagée. Tant que le message CONNECT n'est pas arrivé au TE appelant, il n'y a aucun trafic possible. Il y a cependant certains types d'application critique, telle que la téléphonie, où l'utilisateur signalerait l'acceptation de l'appel par le fait de décrocher le téléphone. L'utilisateur s'attend donc à pouvoir parler directement après avoir décroché. Or, tant que le message signalant l'acceptation n'est pas parvenu jusqu'au TE appelé, aucune ressource réservée ne peut être utilisée. La notion de susceptibilité au *clipping* permet ce que l'on pourrait appeler une ouverture rapide de connexion dans laquelle la ressource réservée (la largeur de bande) serait directement engagée dès sa réservation.

VI- Le descripteur de trafic ATM

L'IE "Descripteur de trafic ATM" est utilisé afin de définir le type de trafic qui sera généré par l'utilisateur. Il permet de spécifier les paramètres que nous avons vus au chapitre 1, section "Descripteur de trafic de la connexion".

Sa structure est exposée aux Figure D-9 et Figure D-10.

Il n'est pas obligatoire de compléter tous les champs de cet IE. Cependant le réseau s'attend à avoir au moins une spécification pour le forward PCR (CLP=1) et/ou backward PCR (CLP=1). Si un de ces éléments n'est pas codé, cela générera une erreur.

On trouvera également à la Figure D-10 un indicateur pour la procédure "Best Effort", exposée au chapitre 1 section "QoS", permettant de travailler avec une QoS non spécifiée. Si l'utilisateur a choisi l'option Best Effort, le réseau s'attend à recevoir également dans cet IE un forward PCR (CLP=1) et/ou un backward PCR (CLP=1).

Les champs "marque en avant" et "marque en arrière" de la Figure D-10 sont utilisés pour indiquer (au choix de l'utilisateur) s'il faut marquer les cellules de priorité CLP=0 dépassant le contrat de trafic spécifié (ou subissant un problème technique du réseau) en priorité CLP=1. Dans ce cas, les cellules ne seront pas effacées (pour peu que le réseau puisse envoyer ces cellules sur le réseau).

Bit	8	7	6	5	4	3	2	1
	0	0	0	0	1	0	0	0
	1	0	0	Flag	Rsvd	Indicateur d'action		
	Longueur de l'IE							
	Longueur de l'IE							
	1	0	0	0	0	0	1	0
	Forward PCR Identifier (CLP=0)							
	Forward PCR							
	Forward PCR							
	Forward PCR							
	1	0	0	0	0	0	1	1
	Backward PCR Identifier (CLP=0)							
	Backward PCR							
	Backward PCR							
	Backward PCR							
	1	0	0	0	0	1	0	0
	Forward PCR Identifier (CLP=1)							
	Forward PCR							
	Forward PCR							
	Forward PCR							
	1	0	0	0	0	1	0	1
	Backward PCR Identifier (CLP=1)							
	Backward PCR							
	Backward PCR							
	Backward PCR							
	1	0	0	0	1	0	0	0
	Forward SCR Identifier (CLP=0)							
	Forward SCR							
	Forward SCR							
	Forward SCR							
	1	0	0	0	1	0	0	1
	Backward SCR Identifier (CLP=0)							
	Backward SCR							
	Backward SCR							
	Backward SCR							

Figure D-9 : structure de l'IE "Descripteur de trafic ATM" (1)

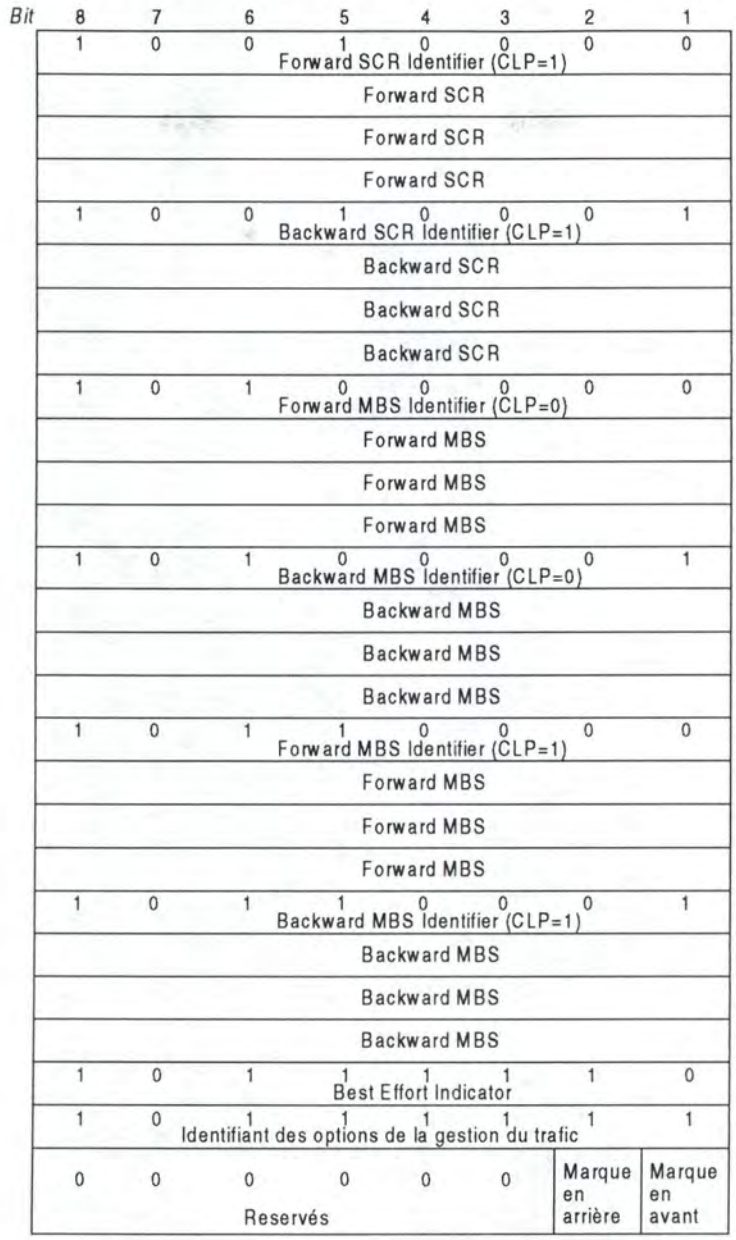


Figure D-10 : structure de l'IE "Descripteur de trafic ATM" (2)

Annexe E : Timers associés à UNI 3.1

Cette annexe reprend l'ensemble des timers associés aux procédures de signalisation du protocole UNI 3.1 de l'ATM Forum.

Dans une première section, nous exposerons les timers se trouvant du côté réseau de l'interface UNI (le point d'accès). Une seconde section exposera les timers se trouvant du côté utilisateur (ATM TE) de l'interface UNI.

1) Timers du côté réseau

Le Tableau E-1 reprend l'ensemble des timers, leur valeur par défaut, l'état de la procédure de connexion lors du déclenchement du timer, la cause de leur démarrage, la condition normale d'arrêt et les actions à mener lors de la première et deuxième expiration.

Nom du timer	*Valeur par défaut*	*Etat de la procédure de connexion*	*Cause du démarrage*	*Arrêt normal*	*1ère expiration*	*2ème expiration*
T303	4 secondes	Call present	Envoi de SETUP	Réception de : CONNECT, CALL PROCEEDING, RELEASE COMPLETE	Retransmission de SETUP et redémarrage de T303	Fermeture de la connexion avec le réseau Passage à l'état *null*.
T308	30 secondes	Release Indication	Envoi de RELEASE	Réception de : RELEASE ou RELEASE COMPLETE	Retransmission de RELEASE et redémarrage de T308	Suppression de la référence d'appel Passage àl'état *null*
T309	10 secondes	Tout état	Déconnexion SAAL (les procédures d'appel en état actif U10-N10 ne sont pas perdues)	SAAL reconnecté	Fermeture de la connexion avec le réseau Suppression de la référence d'appel et du VC	
T310	10 secondes	Incoming call proceeding	Réception du CALL PROCEEDING	Réception de CONNECT ou RELEASE	Fermeture de la connexion par procédure normale (envoi de RELEASE, ...)	
T316	2 minutes	Restart Request	Envoi du message RESTART	Réception du RESTART ACKNOWLEDGE	Retransmission du RESTART	Retransmission du RESTART
T317	implémentation libre mais < T316	Restart	Réception du RESTART	Suppression des références d'appel		
T322	4 secondes	Tout état	Envoi d'un STATUS ENQUIRY	Réception d'un STATUS, RELEASE ou RELEASE COMPLETE	Retransmission du STATUS ENQUIRY	Retransmission du STATUS ENQUIRY
T398	4 secondes	Drop party initiated	Envoi d'un DROP PARTY	Réception d'un DROP PARTY ACK ou d'un RELEASE	Envoi d'un DROP PARTY ACK ou d'un RELEASE	Timer non redémarré
T399	14 secondes	Add party initiated	Envoi d'un ADD PARTY	Réception d'un ADD PARTY ACK, ADD PARTY REJECT ou RELEASE	Suppression de la feuille et passage dans l'état *null* (pour la FSM *party*)	Timer non redémarré

Tableau E-1 : liste et définition des timers pour le côté réseau de l'interface UNI

2) Timers du côté utilisateur

Le Tableau E-2 reprend l'ensemble des timers, leur valeur par défaut, l'état de la procédure de connexion lors du déclenchement du timer, la cause de leur démarrage, la condition normale d'arrêt et les actions à mener lors de la première et deuxième expiration.

Nom du timer	*Valeur par défaut*	*Etat de la procédure de connexion*	*Cause du démarrage*	*Arrêt normal*	*1ère expiration*	*2ème expiration*
T303	4 secondes	Call initiated	Envoi de SETUP	Réception de : CONNECT, CALL PROCEEDING, RELEASE COMPLETE	Retransmission de SETUP et redémarrage de T303	Fermeture de la connexion Passage à l'état *null*.
T308	30 secondes	Release Request	Envoi de RELEASE	Réception de : RELEASE ou RELEASE COMPLETE	Retransmission de RELEASE et redémarrage de T308	Suppression de la référence d'appel Passage à l'état *null*
T309	10 secondes	Tout état	Déconnexion SAAL (les procédures d'appel en état actif U10-N10 ne sont pas perdues)	SAAL reconnecté	Fermeture de la connexion Suppression de la référence d'appel et du VC	
T310	10 secondes	Outgoing call proceeding	Réception du CALL PROCEEDING	Réception de CONNECT ou RELEASE	Fermeture de la connexion par procédure normale (envoi de RELEASE, ...)	
T313	4 secondes	Connect request	Envoi d'un CONNECT	Réception d'un CONNECT ACKNOWLEDGE	Envoi d'un RELEASE	
T316	2 minutes	Restart Request	Envoi du message RESTART	Réception du RESTART ACKNOWLEDGE	Retransmission du RESTART	Retransmission du RESTART
T317	implémentation libre mais < T316	Restart	Réception du RESTART	Suppression des références d'appel		
T322	4 secondes	Tout état	Envoi d'un STATUS ENQUIRY	Réception d'un STATUS, RELEASE ou RELEASE COMPLETE	Retransmission du STATUS ENQUIRY	Retransmission du STATUS ENQUIRY
T398	4 secondes	Drop party initiated	Envoi d'un DROP PARTY	Réception d'un DROP PARTY ACK ou d'un RELEASE	Envoi d'un DROP PARTY ACK ou d'un RELEASE	Timer non redémarré
T399	14 secondes	Add party initiated	Envoi d'un ADD PARTY	Réception d'un ADD PARTY ACK, ADD PARTY REJECT ou RELEASE	Suppression de la feuille et passage dans l'état *null* (pour la FSM *party*)	Timer non redémarré

Tableau E-2 : liste et définition des timers pour le côté utilisateur de l'interface UNI

Annexe F : Traitement des piles de DTL

La Figure F-1 et la Figure F-2 donne l'ensemble de l'algorithme qui doit être appliqué par un noeud exécutant le protocole PNNI lorsque celui-ci reçoit un message SETUP contenant une pile de DTL.

Dans ces figures :

- la notation TP(DTL) signifie « les informations (nodeId et portID) pointées par le Transit Pointer dans la DTL considérée »;
- la notation TP(DTL) :nodeID signifie « l'identifiant du noeud (nodeID) pointé par le Transit Pointer dans la DTL considérée »;
- la notation TP(DTL) :portID signifie « l'identifiant du port (portID) pointé par le Transit Pointer dans la DTL considérée »;
- la notation TopDTL fait référence à la DTL se trouvant en sommet de pile;
- current_node, current_port, output_port et next_destination sont des variables de travail de l'algorithme de traitement des DTL.

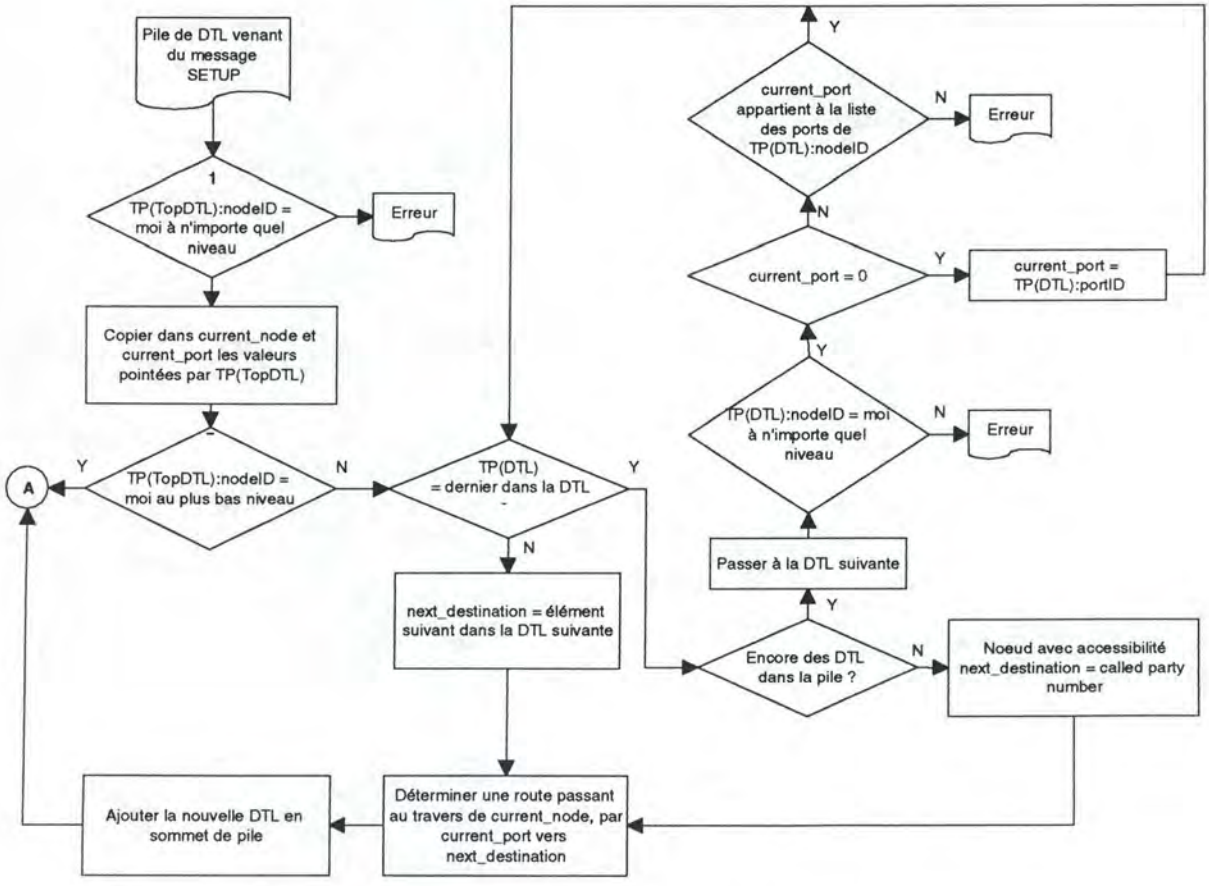


Figure F-1 : algorithme de traitement des DTL (1)

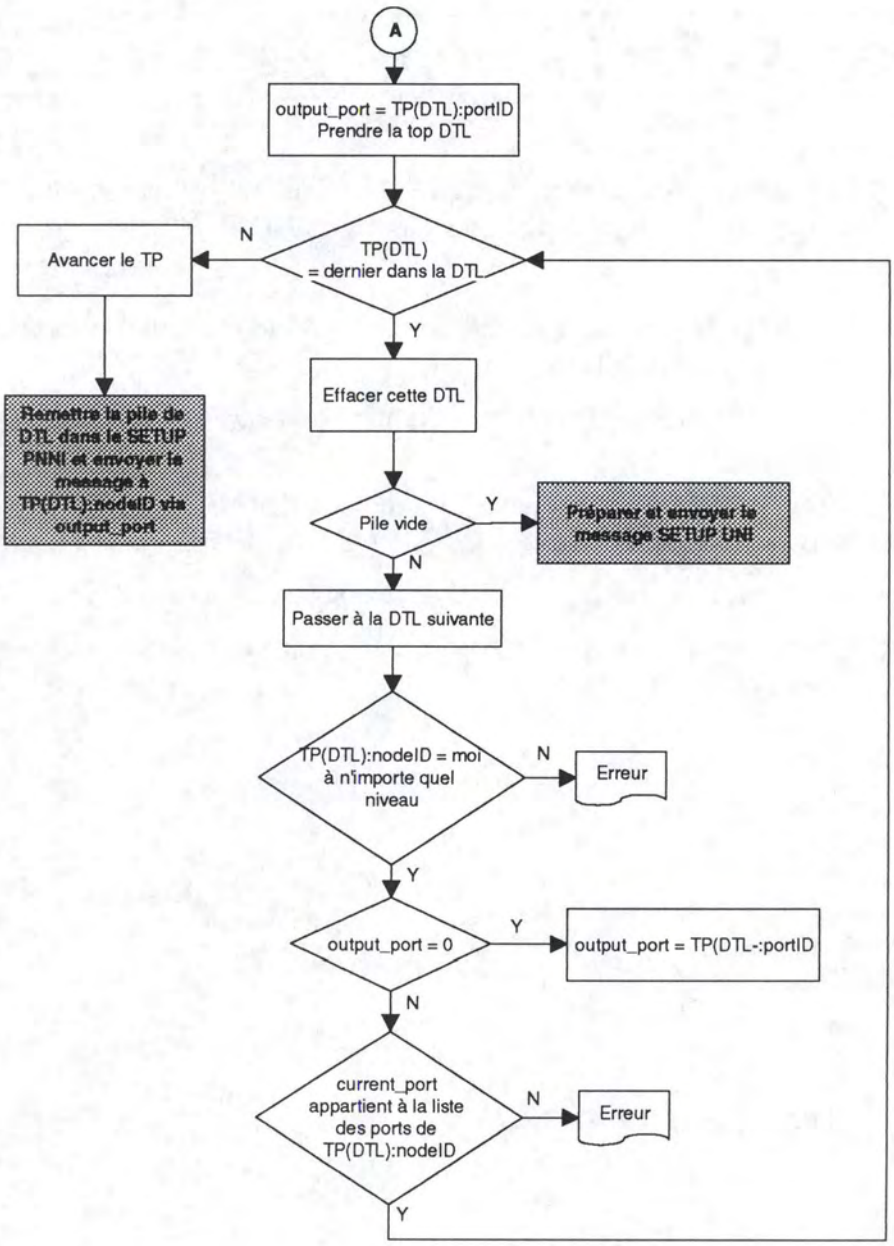


Figure F-2 : algorithme de traitement des DTL (2)

Annexe G : IE particuliers pour PNNI

Il y a deux Information Elements spécifiques au protocole PNNI : l'IE destiné au transport des DTL et l'IE destiné à l'indication de procédures de crankback.

1) L'IE DTL

L'IE utilisé pour le codage des DTL est illustré à la Figure G-1. On retrouve le même codage que celui utilisé en UNI 3.1, c'est-à-dire un premier octet indiquant le type d'IE, un deuxième octet d'indication d'action (tout comme pour UNI 3.1, il indique ce qu'il y a lieu de faire si l'IE reçu n'est pas reconnu) ainsi que 2 octets de codage de la longueur totale de l'IE en dehors des 4 premiers octets.

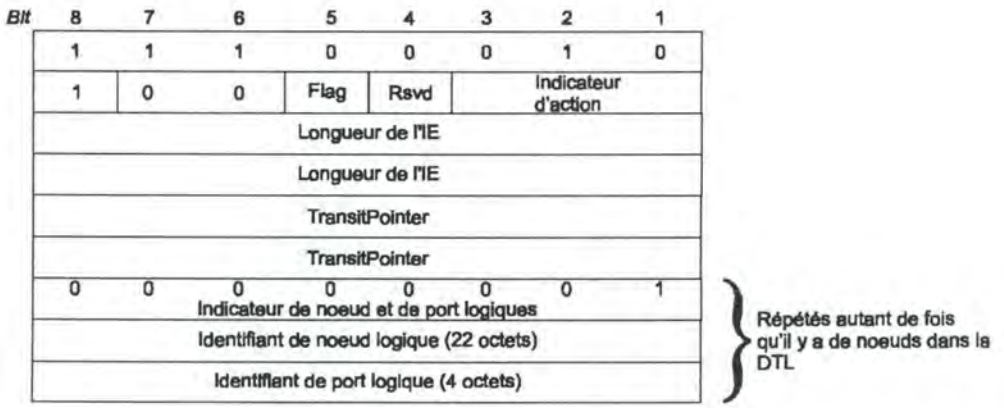


Figure G-1 : structure de l'IE DTL

L'IE DTL se retrouve uniquement dans les messages SETUP et ADD PARTY. Pour chaque DTL de la pile de DTL, il y a un et un seul IE.

L'indicateur de noeud et port logique est utilisé afin de séparer tous les couples identifiant de noeud / identifiant de port constituant une DTL. Les deux octets « Transit Pointer » sont utilisés afin d'encoder l'index du noeud sur lequel on se trouve actuellement (voir les exemples de procédures de signalisation donnés au chapitre 5).

Lorsqu'une pile de DTL est composée de plusieurs DTL, on commence toujours par encoder dans le message SETUP ou ADD PARTY l'IE DTL correspondant au plus haut niveau de la hiérarchie pour terminer par l'IE représentant le plus haut niveau de la hiérarchie.

2) L'IE crankback

Lors d'une procédure de crankback, l'indication de cette procédure est signalée par la présence d'un IE crankback dans un message RELEASE, RELEASE COMPLETE ou ADD PARTY REJECT.

Cet IE est structuré selon la Figure G-2.

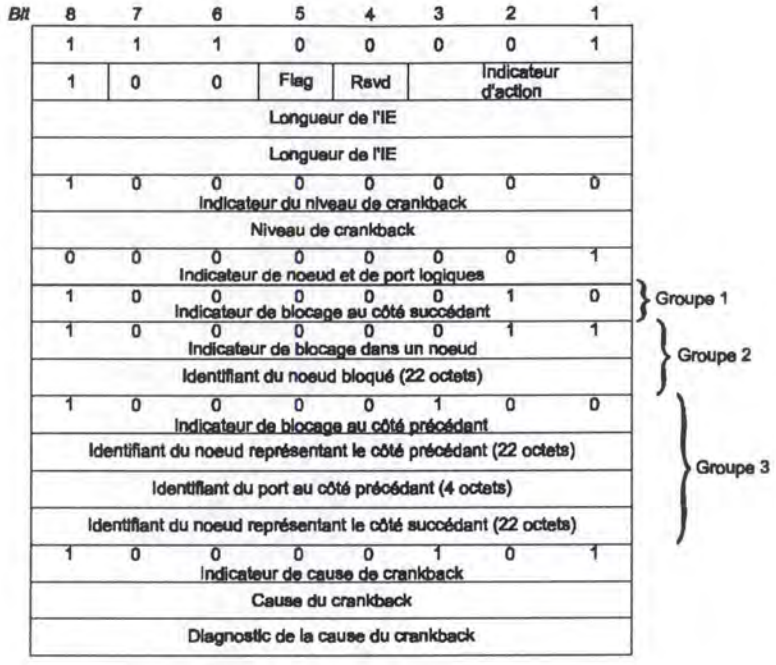


Figure G-2 : structure de l'IE crankback

Reprenons les différents champs de cette structure :

- *l'indicateur de niveau de crankback* : l'indicateur de niveau de crankback sert à encoder le niveau d'une DTL de sommet de pile. Le niveau d'une DTL est le niveau de tous les noeuds qui la composent. Lors d'un établissement de connexion, si une procédure de crankback doit être démarrée, on encodera dans l'indicateur de niveau de crankback le niveau de la DTL de sommet de pile que l'on était en train de parcourir;
- *l'indicateur de blocage au côté succédant* : lorsqu'un blocage au côté succédant d'un lien est détecté, un IE crankback est inclus dans le message RELEASE ou ADD PARTY REJECT. L'indicateur de blocage au côté succédant est alors la seule information donnée au sujet du crankback;
- *l'indicateur de blocage dans un noeud* : lorsqu'un blocage dans un noeud est détecté on identifiera la source du blocage dans l'IE crankback par cet indicateur et l'identifiant du noeud qui a causé le blocage;
- *l'indicateur de blocage au côté précédant* : lorsqu'un blocage au côté précédant d'un lien est détecté on identifiera la source du blocage dans l'IE crankback par cet indicateur. Il sera complété par l'identifiant du noeud représentant le côté précédant du lien, par l'identifiant du port correspondant à ce lien dans le noeud identifié ainsi que par l'identifiant du noeud représentant le côté succédant du lien. Ces trois valeurs permettent d'identifier sans équivoque le lien posant problème;
- *la cause de crankback* : nous avons vu que lorsqu'un message de type RELEASE était généré, celui-ci incluait un IE Cause permettant d'identifier la raison de l'émission de ce message. De même, une cause sera spécifiée dans un IE crankback afin de permettre de déterminer les raisons pour lesquelles la procédure de crankback a été amorcée. L'indicateur de cause permet « d'annoncer » la valeur de la cause codée dans l'octet qui suit. L'octet de diagnostic permet quant à lui, tout comme pour l'IE Cause de UNI 3.1, d'apporter des informations complémentaires quant à la cause spécifiée.

Les groupes 1, 2 et 3 de la Figure G-2 sont mutuellement exclusifs.

Annexe H : Allocation des VCI et VPI dans PNNI

Nous avons donné dans le chapitre consacré à la signalisation entre noeuds d'un réseau ATM privé une méthode d'allocation d'un couple de valeurs VPI/VCI. Cette méthode, identique à celle utilisée dans le protocole UNI, consistait simplement à allouer tout couple de valeurs VPI et VCI libre, en dehors des valeurs réservées pour certains protocoles[1].

A l'encontre de UNI 3.1, il est possible dans le protocole PNNI d'utiliser deux méthodes distinctes d'allocation des couples VPI/VCI. Il s'agit de la signalisation associée (*associated signaling*) et de la signalisation non-associée (*non-associated signaling*).

1) Signalisation associée

En signalisation associée, le côté précédant d'un lien demande explicitement au côté succédant du lien d'allouer un canal dans le VP contenant le VC de signalisation. Le VP doit alors explicitement être indiqué dans le message SETUP reçu par le côté succédant du lien. Ceci implique donc la présence obligatoire de l'IE "identifiant de connexion" tel que présenté à l'annexe D, section IV. Si cet IE n'est pas présent dans le message SETUP reçu, on se trouve donc obligatoirement dans un cas de signalisation non-associée. De plus, l'IE "identifiant de connexion", s'il est présent, doit spécifier explicitement qu'il s'agit d'un cas de signalisation associée (un champ est réservé à cet usage dans l'IE).

On distingue deux méthodes d'allocation en signalisation associée selon que l'on spécifie dans l'IE "identifiant de connexion" :

1. VPI exclusif et tout VCI : dans ce cas, le côté succédant du lien choisit tout VCI disponible dans le VPI spécifié (qui est obligatoirement le VPI du VP contenant le VC de signalisation). Si aucune valeur ne peut être allouée, l'appel est rejeté et une procédure de crankback doit être démarée.
2. VPI exclusif et VCI exclusif : dans ce cas, le côté précédant du lien a spécifié dans l'IE "identifiant de connexion" contenu dans le message SETUP envoyé un VCI qu'il désire utiliser. Si cette valeur est disponible au côté succédant du lien, celle-ci est sélectionnée et le message CALL PROCEEDING envoyé en acquittement au côté précédant du lien contient un IE "identifiant de connexion" spécifiant les mêmes valeurs VPI/VCI que celles contenues dans le messages SETUP reçu. Si cependant la valeur VCI spécifiée dans le message SETUP reçu ne peut être allouée au côté succédant du lien, l'appel (i.e. la demande de connexion) est refusé et une procédure de crankback doit être démarrée.

[1] Rappelons-nous que pour PNNI par exemple, le couple (VPI=0 ; VCI=5) était réservé au module de signalisation. En fait, les valeurs comprises entre 0 et 31 ne peuvent être allouées à un VCI pour une demande d'ouverture de connexion (les valeurs de 0 à 15 sont réservées pour l'ITU-T, celles de 16 à 31 sont réservées pour l'ATM Forum). On choisira alors une valeur entre 32 et 65535 pour cet identifiant. Il n'y a pas de limitations pour l'allocation d'un VPI.

2) Signalisation non-associée

La signalisation non-associée doit être envisagée dans trois cas :

1. le message SETUP reçu par le côté succédant du lien ne contient pas d'IE "identifiant de connexion". Dans ce cas, le côté succédant choisit toute valeur VPI/VCI disponible. Si aucune valeur ne peut être allouée, l'appel est rejeté et une procédure de crankback doit être démarrée.
2. le message SETUP reçu par le côté succédant contient un IE "identifiant de connexion" spécifiant une signalisation non-associée. De plus, cet IE contient une indication "VPI exclusif et tout VCI". Dans ce cas, l'IE spécifie une valeur VPI qui ne doit pas, comme c'était le cas dans la signalisation associée, être obligatoirement égale au VPI du VP contenant le VC de signalisation. Si ce VPI est disponible et que l'on peut trouver une valeur VCI libre pour celui-ci, l'appel est accepté et le message CALL PROCEEDING envoyé en acquittement spécifie le couple VPI/VCI défini. Si le VPI n'est pas disponible ou que l'on ne peut trouver un VCI libre pour celui-ci, l'appel est rejeté et une procédure de crankback doit être démarrée.
3. le message SETUP reçu par le côté succédant contient un IE "identifiant de connexion" spécifiant une signalisation non-associée. De plus, cet IE contient une indication "VPI exclusif et VCI exclusif". Ce cas diffère du précédent par l'indication, dans le message SETUP, d'un couple VPI et VCI que le côté succédant du lien désire utiliser. Si le VPI ou le VCI spécifié n'est pas disponible, l'appel est rejeté et une procédure de crankback doit être démarrée.